Digitized by Sarayu Trust Foundation Delhi and eGangotri.

Development of Sanskrit Computational Toolkit and Sanskrit-Hindi Machine Translation System (SHMT)

Project Report

January, 2011

Funded By
Department of Information Technology
Ministry of Communication and Information Technology
India

PUBLICATIONS

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

PAPERS

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

INDEX

S. NO.	PAPER	PAGE NO.
1	Annotating Sanskrit Texts based on Sabdabodha systems	1
2	Annotating Sanskrit corpus : adapting IL-POSTS	13
3	Automatic Sanskrit Segmentizer Using Finite State Transducers	19
4	Designing a Constraint Based Parser for Sanskrit in **English**	25
5	Designing a Constraint Based Parser for Sanskrit in **Sanskrit**	47
6	Evaluating Tagsets for Sanskrit	69
7	Sanskrit Morphological Analyser	83
8	Sanskrit Compound Processor in **English**	95
9	Sanskrit Compound Processor in **Sanskrit**	109
10	Sanskrit Compound Paraphrase Generator	125
11	Vibhakti Divergence between Sanskrit and Hindi in **English**	135
12	Vibhakti Divergence between Sanskrit and Hindi in **Sanskrit**	147
13	About in the Project in **Sanskrit**	159

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

Annotating Sanskrit Texts based on Śābdabodha systems

K. V. Ramkrishnamacharyulu

Rastriya Sanskrit Vidyapeetham, Tirupati, India

1 वन्दनम्

श्री रामो विजयतेतराम्। श्री रामाय नमः। श्री गुरुभ्यो नमः। श्रीमद्रामानुजार्य नमः।

2 Introduction

Though Sanskrit has a huge repository of texts as well as well discussed grammar formalism, we still neither have a full fledged parser for Sanskrit based on this formalism nor do we have any annotated text.

We propose here a tagging scheme for manual tagging of Sanskrit texts, based on different grammatical relations that have been discussed by different schools of śābdabodha systems. We hope the tagging scheme proposed here serves as a good starting point for manual annotation.

The process of Śābdabodha involves identifying the relations between different words in a sentence. The traditional model of sentence analysis which is mainly based on Pāṇinian model provides us with various relations that are mainly Syntactico-Semantic in nature. Tradition treats a sentence as a series of modifier-modified relations(M.M.R.). *Ākāṅkṣā* (expectancy) plays a major role in establishing these relations.

Sentence is a group of words that are bound together by ākāṅkṣā. Sentences are of two types: vākya (simple sentence) and mahā-vākya (complex sentence).

Vākya is a group of words with one verb. Other words in the sentence satisfy the ākāṅkṣās (expectancies) of the main verb. These ākāṅkṣās are called utthita ākāṅkṣā(Natural expectancies). The main verb is the mukhya viśeṣya (head) which is modified by other words (modifiers) in the sentence.

Mahā vākya is a group of sentences which are inter-related and denote a single meaning. Here the head of the main sentence is modified by the heads of the other (subordinate) sentences. As such the relations between the main verb of the main sentence and the main verbs in sub-ordinate sentences denote

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

the inter-sentential relations. The ākāṅkṣā between the activities i.e. heads of the sentences is not natural but are triggered by special words in the sentence construction. These ākāṅkṣās, therefore, are called *utthāpyākāṅkṣā*.

The relations, thus, are of two types: external relations - relations between sentences and internal relations - relations between words within the sentences.

In what follows, we list different inter-sentential and intra-sentential relations. We indicate, wherever possible, the marker which helps in identifying these relations, semantics associated with these relations, language conventions associated with them, if any, followed by the proposed tag[1] with an example sentence.

3 Proposed Tagset

3.1 Inter-sentential relations

These relations are denoted by certain link words and non-finite verb forms. The ākāṅkṣā between two sentences is known only if the link words are present in the sentence. All such relations are marked by (...)_R where (...) denotes the part of the sentence, R is the inter-sentential relation. The inter-sentential relations may be further subclassified as

1. Relations denoted by non-finite verbs
2. Relations denoted by certain link words

Relations denoted by Non-finite verbs

1. पूर्वकालीनत्वम्
   liṅgam: ktvā ending non-finite verb form
   Meaning: ktvā- ending verb form denotes the activity which preceeds an activity denoted by the main verb.
   Properties: The kartā of the ktvā ending verb is shared with that of the main verb[2]
   Proposed tag: (...)_ktvā
   Example: (रामः दुग्धं पीत्वा)_ktvā शालां गच्छति ।

[1] Though technically name of the relation should end in the bhāva pratyaya 'tva' as in 'kartṛtva', 'karmatva' etc., since we tag the words, we tag them as 'kartā', 'karma', etc.
Thus in the sentence
रामः गृहं गच्छति।
rāma has kart.rtva of the kriyā denoted by *gam*, but we tag it as
रामः_kartā गृहं गच्छति ।

[2] समानकर्तृकयोः पूर्वकाले 3.4.21

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

2. प्रयोजनम् (purpose of the main activity)
lingam: tumun-ending nonfinite verb form
Meaning: The tumun-ending verb denotes the purpose of the main activity
Properties: The kartā is shared in some cases. In some cases the karma or sampradāna of the main activity becomes the kartā of the purpose activity.
proposed tag: (...)_tumun
Example: (अहं प्रतिदिनं (योगशास्त्रं पठितुं)_tumun विद्यालयं गच्छामि)।
(अहं भवन्तं मम गृहे (भोक्तुम्)_tumun आह्वयामि) ।

3. समकालीनत्वम् (Simultaneity)
lingam: śatṛ or śānac
Meaning: It denotes the activity occuring simultaneously with the main verb
Properties: The kartā is shared with the main activity
proposed tag: (...)_śatṛ/(...)_śānac
Example: (बालकः (जलं पिबन्)_śatṛ गच्छति) ।
(बालकः (शयानः)_śānac हसति) ।

4. भावलक्षणसप्तमी

– अनन्तरकालिकत्वम् (Time of the completion of preceeding activity)
lingam: kta - ending in 7^{th} case
proposed tag: (...)_kta_7
Example: ((रामे वनं गते)_kta_7 दशरथः खिन्नः) ।

– समकालीनत्वम् (simultaneous events)
lingam: śatṛ or śānac - ending in 7^{th} case
propsed tag: (...)_śatṛ$_7$
Example: ((रामे वनं गच्छति)_śatṛ$_7$ सीता अनुसरति) ।

– पूर्वकालीनत्वम् (time of the main activity before the starting of the subordinate activity)
lingam: lṛṭ + śatṛ or śānac ending in 7^{th} case
Proposed tag: (...)_lṛṭ-śatṛ$_7$
Example: ((गोषु धोक्ष्यमाणासु)_lṛṭ-śatṛ$_7$ गतः) ।

Relations denoted by words

1. समानकालीनत्वम्

– lingam: yadā - tadā or yasmin kāle - tasmin kāle
yadā or yasmin kāle in the subordinate sentence typically in the beginning and tadā or tasmin kāle in the beginning of the main sentence
proposed tag: (...)_yadā$_1$

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

Example: (तदा मयूरः नृत्यति (यदा मेघः वर्षति) $yadā_1$)।

– liṅgam: yadā or yasmin kāle
Only yadā or yasmin kāle is present, and tadā or tasmin kāle is absent.
proposed tag: (...)_$yadā_2$
Example: (मयूरः नृत्यति (यदा मेघः वर्षति)_$yadā_2$) ।

– liṅgam: tadā or tasmin kāle
Only tadā or tasmin kāle is present, and yadā or yasmin kāle is absent.
proposed tag: (...)_tadā
Example: (मेघः वर्षति (तदा मयूरः नृत्यति)_tadā) ।

2. प्रतिबन्धः (conditional relation)

– liṅgam: yadi - tarhi
'yadi' in the beginning of a subordinate sentence and 'tarhi' in the main sentence;
proposed tag: (...)_$yadi_1$
Example: ((यदि त्वम् इच्छसि)_$yadi_1$ तर्हि अहं भवतः गृहम् आगच्छामि) ।

– liṅgam: yadi
Only yadi is used,
proposed tag: (...)_$yadi_2$
Example: (अहम् आगमिष्यामि (यदि भवान् अपेक्षितं सौलभ्यं विधास्यति)_$yadi_2$) ।

– liṅgam: tarhi
Only 'tarhi' is used, and the word 'yadi' is missing. proposed tag (...)_tarhi
Example: (त्वम् इच्छसि (तर्हि अहं भवतः गृहम् आगच्छामि)_tarhi) ।

– liṅgam: cet
presence of the word 'cet' proposed tag: (...)_cet
Example: ((त्वम् इच्छसि चेत्)_cet अहं भवतः गृहम् आगच्छामि) ।

– liṅgam: tarhi eva
group the words from the beginning up to tarhi_eva as one sentence, and the rest as second sentence
proposed tag: (...)_tarhi_eva
Example: ((त्वम् इच्छसि तर्हि एव)_tarhi_eva अहं भवतः गृहम् आगच्छामि) ।

3. कारणसत्त्वेऽपि कार्याभावः, कारणाभावेऽपि कार्योत्पत्तिः (Non productive effort (or) product without cause)

– liṅgam: yadyapi – tathāpi
proposed tag (...)_$yadyapi_1$
Example: ((यद्यपि अयं बहु प्रयासं कृतवान्)_$yadyapi_1$ तथापि परीक्षा तु

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

अनुत्तीर्णा) ।

– liṅgam: yadyapi
Example: ((यद्यपि अनेन बहु प्रयासः कृतः)_yadyapi$_2$ परीक्षा तु अनुत्तीर्णा) ।

– liṅgam: tathāpi proposed tag (...)_tathāpi
Example: (अयं तथा न कुशलः (तथापि प्रथमपुरस्कारं लब्धवान्)_tathāpi) ।

– liṅgam: athāpi or evamapi
proposed tag (...)_athāpi
Example: (ज्वरः अपगतः (अथापि औषधं सेव्यते)_athāpi)

4. हेतुहेतुमद्भावः (cause and effect)

– liṅgam: yataḥ-tataḥ or yasmāt-tasmāt
proposed tag (...)_yataḥ$_1$
Example: ((यतः अयं समये नागतः)_yataḥ1 ततः प्रवेशपरीक्षायां नानुमतः) ।

– liṅgam: yataḥ or yasmāt
proposed tag (...)_yataḥ$_2$
Example: (प्रवेशपरीक्षायां नानुमतः अयं (यतः समये नागतः)_yataḥ$_2$) ।

– liṅgam: tataḥ or tasmāt or ataḥ
proposed tag (...)_tataḥ
Example: (अयं समये नागतः (ततः अयं परीक्षायां नानुमतः)_tataḥ)।

5. अनन्तरकालीनत्वम् (following action)
liṅgam: tataḥ or tatastataḥ or anantaram or atha
proposed tag: (...)_atha
Example: (प्रथमम् अहं शृणोमि (अथ लिखामि)_atha) ।

6. समुच्चयः (conjunction)
liṅgam: api_ca or kim_ca
proposed tag (...)_apica
Example: (भिक्षाम् अट (अपिच गामानय)_apica) ।

7. समानाधिकरणत्वम् (co-location)
– liṅgam: yatra - tatra or yasmin - tasmin
proposed tag: (...)_yatra$_1$
Example: ((यत्र नार्यस्तु पूज्यन्ते)_yatra$_1$ रमन्ते तत्र देवताः) ।

– liṅgam: yatra or yasmin
proposed tag: (...)_yatra$_2$
(अहो बृन्दावनं रम्यं (यत्र गोवर्धनो गिरिः)_yatra$_2$) ।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

– liṅgam: tatra or tasmin
proposed tag: (...)_tatra
Example: ((तत्र स्नात्वा नरो राजन्)_tatra गोसहस्रफलं लभेत) ।

8. असाफल्यम् (non-fulfilment of expected activity)
liṅgam: kintu or parantu
proposed tag (...)_kintu
Example: (गजेन्द्रः तीव्रप्रयत्नम् अकरोत् (किन्तु नक्रग्रहात् न मुक्तः)_kintu) ।

4 Sentence Internal relations

These are of two types

– related to the words denoting activity,
– related to other words.

4.1 Relations related to the activity-denoting words

These relations are triggered by the vibhaktis. However one vibhakti may indicate several relations. It is the context which indicates a particular relation. we mark these relations by REL where REL stands for the relation label.
These relations are also of two types:

– kāraka relations
– non-kāraka relations

– kāraka relations
  - kartā k_1
  - karma k_2
  - karaṇa k_3
  - sampradāna k_4
  - apādāna k_5
  - adhikaraṇa k_7

These kāraka relations may also be further classified as

- कर्ता (kartā) (k_1)
  * default (k_1)
    देवदत्तः_k_1 पचति।
    रथः_k_1 गच्छति।
  * अनुभवी (experiencer(k_1-e)
    Example: घटो_k_1-e नश्यति।
    पुत्रः_k_1-e जायते।
    सः_k_1-e सुखम् अनुभवति।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

* अमूर्तः (abstract) (k_1-a)
  Example: क्रोधः_k_1-a आगच्छति।

* प्रयोजकः (prayojakaḥ) (k_1-p)
  देवदत्तः_k_1-p विष्णुमित्रेण पाचयति।

* प्रयोज्यः (prayojyaḥ) (k_1-j)
  देवदत्तः विष्णुमित्रेण_k_1-j पाचयति।

* मध्यस्थः (madhyasthaḥ) (k_1-m)
  देवदत्तः यज्ञदत्तेन k_1-m विष्णुमित्रेण पाचयति।

* अभिप्रेरकः/उत्प्रेरकः (cause for temptation) (k_1-t)
  Example: मोदकः_k_1-t रोचते।

* कर्म-कर्तृ (karma-kartṛ) (k_1-k_2)
  Example: भिद्यते काष्ठः_k_1-k_2 स्वयमेव।
  पच्यते ओदनः_k_1-k_2 स्वयमेव।

* करण-कर्तृ (karaṇa-kartṛ) (k_1-k_3)
  Example: अंसिः_k_1-k_3 छिनत्ति।

* षष्ठी-कर्ता (ṣaṣṭhī kartā) (K1-6)
  Example: आचार्यस्य_k_1-6 अनुशासनम्।

- कर्म (karma) (k_2)
  * default (k_2)
    Example: शत्रून्_k_2 जयति।
    ओदनं_k_2 भुङ्क्ते।

  * उत्पाद्यम् (created)(k_2-u)
    Example: ओदनं_k_2-u पचति।

  * विकार्यम् (raw meterial)(k_2-v)
    Example: सुवर्णं_k_2-v कुण्डलं करोति।

  * प्रयोज्य-कर्ता (prayojya-kartā) (k_2-j)
    Example: बालं_k_2-j क्षीरं पाययति।

  * आधारः (location) (k_2-l)
    Example: वैकुण्ठम्_k_2-l अधिशेते।

  * देशः (village, town, state, country etc) (k_2-p)
    Example: कुरून्_k_2-p स्वपिति।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

* कालः (time) (k_2-t)
  Example: मासम्_k_2-t आस्ते।

* भावः (activity) (k_2-a)
  Example: गोदोहम्_k_2-a आस्ते।

* मार्गः (road measurment) (k_2-m)
  Example: क्रोशम्_k_2-m आस्ते।

* सम्प्रदानम् (recipient) (k_2-k_4)
  Example: पशुना रुद्रं_k_2-k_4 यजते।

* अनीप्सितम् (not intended) (k_2-an)
  Example: ग्रामं गच्छन् तृणं_k_2-an स्पृशति।

* अकथितम् (not expected) (k_2-un)
  Example: गोपः गां_k_2-un दोग्धि पयः।

* गति-कर्म (gati-karma) (k_2-g)
  Example: रामः ग्रामं_k_2-g गच्छति।

* करणम् (instruments of playing) (k_2-k_3)
  Example: अक्षान्_k_2-k_3 दीव्यति।
  कन्दुकं_k_2-k_3 क्रीडति।

* यं प्रति कोपः (yaṃ prati kopaḥ) (k_2-k)
  Example: क्रूरम्_k_2-k अभिक्रुध्यति।

* मन्य-कर्म (in disrespect) (k_2-d)
  Example: न त्वां तृणाय_k_2-d / तृणं_k_2-d मन्ये।

* षष्ठी-कर्म (ṣaṣṭhī-karma) (k_2-6)
  Example: शब्दानाम्_k_2-6 अनुशासनम्।

- करणम् (instrument) (k_3)

  * default (k_3)
    बालः कुञ्चिकया तालम् उद्घाटयति।

  * कर्म (karma) (k_3-k_2)
    Example: पशुना_k_3-k_2 रुद्रं यजते।

  * परिक्रयणम् (money in bonded labour) (k_3-m)
    Example: शतेन_k_3-m परिक्रीणाति।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

- सम्प्रदानम् (recipient) (k_4)

  * स्वत्वाश्रयः (recipient with ownership) (k_4-o)
    देवदत्तः ब्राह्मणाय_k_4-o गां ददाति।

  * स्वीकर्ता (recipient without ownership) (k_4)
    देवदत्तः रजकाय_k_4-o वस्त्रं प्रक्षालनाय ददाति।

  * क्रियया अभिप्रेतः (intended to relate with activity) (k_4-i)
    example: पत्ये_k_4-i शेते।

  * ज्ञीप्स्यमानः (addressed through praise etc.) (k_4-a)
    example: कृष्णाय_k_4-a श्लाघते।

  * उत्तमर्णः (a creditor) (k_4-u)
    example: देवदत्ताय_k_4-u शतं धारयति।

  * ईप्सितम् (desired) (k_4-d)
    Example: पुष्पेभ्यः_k_4-d स्पृहयति।

  * यं प्रति कोपः सः (point of anger) (k_4-k)
    Example: हरये_k_4-k क्रुध्यति।

  * प्रीयमाणः (location of desire) (k_4-p)
    Example: देवदत्ताय_k_4-p रोचते मोदकः।

  * यस्य विप्रश्नः (enquiry about) (k_4-e)
    Example:- कृष्णाय_k_4-e राध्यति।

  * परिक्रयणम् (money in bonded labour) (k_4-b)
    Example: देवदत्तः शताय_k_4-b परिक्रीतः।

- अपादानम् (apādānam) (k_5)
  * default (point of departure/seperation) (k_5)
    वृक्षात्_k_5 पर्णं पतति।

  * भय-हेतुः (cause of fear) (k_5-f)
    Example: गृहस्थः चोरात्_k_5-f बिभेति।

  * आख्याता-उपयोगे (teacher) (k_5-u)
    Example: छात्रः उपाध्यायात्_k_5-u अधीते।

  * यस्मात् वारणम्(point for obstruction) (k_5-o)
    Example: कूपात्_k_5-o अन्धं वारयति।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

* यस्य / यस्या अदर्शनम् इष्टं सः / सा (person intended not to be seen) (k_5-n)
  Example: मातुः_k_5-n निलीयते कृष्णः।

* प्रकृतिः (raw material) (k_5-p)
  Example: मृदः_k_5-p घटः जायते।

* प्रभवः (place of first appearence) (k_5-a)
  Example: हिमवतः_k_5-a गङ्गा प्रभवति।

* पराजयः (defeat from activity)[3] (k_5-d)
  Example: अध्ययनात्_k_5-d पराजयते।

- अधिकरणम् (location) (k_7)
  * कालः (time) (k_7-t)
    Example: त्रेतायुगे_k_7-t रामः आसीत्।

  * देशः deśaḥ (place) (k_7-p)
    Example: रामः आयोध्यायाम्_k_7-p आसीत्।

  * विषयः viṣayaḥ (other than above) (k_7-v)
    Example: मोक्षे_k_7-v इच्छा अस्ति।

  * समयस्य अवधिः (time duration) (k_7-td)
    Example: जनवरीतः_k_5-a (मई पर्यन्तं)_k_7-td विरामः।

  * अन्तराल-देशः (place in between) (k_7-pd)
    Example: तिरुपतितः_k_5-a चन्द्रगिरिपर्यन्तं_k_7-td भवनानि सन्ति।

– अकारकसम्बन्धः साक्षात् क्रियया (Non kāraka relations, but direct relations with the activity)

- सम्बोधनम् (addressed) (radr)
  Example: भो राम_radr माम् उद्धर।

- प्रसज्यप्रतिषेधः (uncompounded negation) (rneg)
  Example: रामः गृहं न_rneg गच्छति।

- साम्यम् (similarity) (rs)
  Example: ब्राह्मणवत्_rs अधीते।

- क्रिया - आवृत्त्यन्तरालसमयः (time duration between the repetition of the same activity) (rtd)

[3] पराजेः असोढः parājeḥ asoḍhaḥ 1.4.26

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

Example: अद्य भुक्त्वा दिनद्वयात्_rtd भोक्ता।

- तादर्थ्य (purpose) (rtv)
  Example: छात्रः अध्ययनाय_rtv विद्यालये वसति।
  सा क्रयणाय_rtv आपणं गच्छति

- हेतुः (cause) (rhv)
  Example: विद्यार्थी अध्ययनेन_rhv विद्यालये वसति।

- वीप्सा (repitition) (rrpt)
  Example: शकुन्तला आश्रमे प्रतिवृक्षं_rrpt सिञ्चति।

- क्रिया-आवृत्ति-गणना (counting of repetition) (rcrpt)
  Example: बालकः पाठं पञ्चवारं_rcrpt पठति।

- क्रियाविशेषणम् (manner adverb) (rad)
  Example: हस्ती मार्गे मन्दं_rad गच्छति।
  मृगः वेगेन_rad धावति।

- अत्यन्त-सम्बद्धः कालः (complete relation with time) (rt2)
  Example: बालकः गुरुकुले मासम्_rt2 अधीतवान्।

- अत्यन्त-सम्बद्धः मार्गः (complete relation with road) (rr2)
  Example: पाठः क्रोशम्_rr2 अधीतः।

- अत्यन्त-सम्बद्धः कालः सफलः (complete relation with time with result) (rt3)
  Example: बालकेन मासेन_rt3 अनुवाकः अधीतः।

- अत्यन्त-सम्बद्धः मार्गः सफलः (complete relation with road with result) (rr3)
  बालकेन क्रोशेन_rr3 अनुवाकः अधीतः।

– Other Relations
- षष्ठी (ṣaṣṭhī relation) (r6)
  (अध्यापकस्य)_r6 पुस्तकं छात्राः पठन्ति।

- आरम्भसमयः मापने (starting point of time) (rst5)
  Example: कार्तिक्याः_rst5 आग्रहायणी मासे।

- आरम्भदेशः मापने (starting point of place) (rsp5)
  Example: तिरुपतितः_rsp5 चन्द्रगिरिः क्रोशे।

- लक्षणम् (point of direction) (rd)
  Example: वृक्षं प्रति_rd विद्योतते विद्युत्।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

पक्षी भवनस्य उपरि_rd डयते।
रामः ग्रामं प्रति_rd गतः।

- तादर्थ्य (purpose) (rt)
  Example: बालकाय_rta पुस्तकं क्रीणाति।

- हेतुः hetuḥ (rh)
  Example: कुम्भकारः दण्डेन_rh घटं करोति।

- सह सम्बन्धः (associative) (ras)
  (पुत्रेण सह)_ras पिता गच्छति।

- विना (non-associative) (rnas)
  (धर्मेण विना)_rnas जीवनं नास्ति।

- विभक्तः (comparison between two) (rv5)
  माथुराः पाटलीपुत्रकेभ्यः_rv5 आढ्यतराः।

- निर्धारणम् (Isolating one from a group – in the superlative degree context) (rn7 / rn6).
  गवां_rn7 कृष्णा बहुक्षीरा।
  गोषु_rn6 कृष्णा बहुक्षीरा।

5 Appendix

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Linguistic & Technology Conference, Poland Nov 2009
Digitized by Sarayu Trust Foundation Delhi and eGangotri

(13)

Annotating Sanskrit corpus: adapting IL-POSTS

Girish Nath Jha
Special Center for Sanskrit Studies
Jawaharlal Nehru University, New Delhi
girishjha@gmail.com

Madhav Gopal
Center for Linguistics, SLL&CS,
Jawaharlal Nehru University, New Delhi
mgopalt@gmail.com

Diwakar Mishra
Special Center for Sanskrit Studies
Jawaharlal Nehru University, New Delhi
diwakarmishra@gmail.com

Abstract
In this paper we present an experiment on the use of the hierarchical Indic Languages POS Tagset (IL-POSTS) (Baskaran et al 2008 a&b), developed by Microsoft Research India (MSRI) for tagging Indian languages, for annotating Sanskrit corpus. Sanskrit is a language with richer morphology and relatively free word-order. The authors have included and excluded certain tags according to the requirements of the Sanskrit data. A revision to the annotation guidelines done for IL-POSTS is also presented. The authors also present an experiment of training the tagger at MSRI and documenting the results.

1. Introduction

Sanskrit, the oldest classical language of India, is also the oldest documented language of the Indo-European family. *ṛgveda* (1500 BCE) is the oldest text of this family contains a sophisticated use of the pre-Pāṇinian variety also called *vaidikī*. Pāṇini variously calls his mother tongue *bhāṣā* or *laukikī*. His grammar has two sets of rules – for *vaidikī* (variety used in the vedas) and for *laukikī* (variety used by the common people). The term 'Sanskrit' (meaning 'refined') is given to the standard form of *laukikī* (current language) which emerged after Pāṇini's grammar *Aṣṭādhyāyī* (AD) (700 BCE) (Jha et al 2007).

Structurally Sanskrit is relatively free word-order and inflected language with amazing capacity to synthesize new sounds and morphemes at the word/sound junctures. Morphologically very elaborate, the rules of Sanskrit grammar are precisely done by Pāṇini. However, these rules are not always easy to completely solve computationally (Jha et al 2009). The viability and usefulness of POS annotation in Sanskrit has been often doubted with the argument that a good morphological analyzer is what would be needed for it. But resolution of ambiguous labels in Sanskrit, as in other languages, cannot be done by morph analyzer alone.

The relatively large computational linguistics community in India does not have a single standard for annotating linguistic data. There are in fact many 'standards'. Five language families with diversity being a norm than exception, the feasibility of a single framework was considered un-attainable until 2008, when Microsoft Research India initiated a collaborative initiative in developing a common framework for Indian languages based on EAGLES guidelines. The framework which resulted is called IL-POSTS (Indic Languages POS Tag Set) (Baskaran et al 2008 a&b). This framework has since been successfully tried on many Indian languages across many families. The present research is a report on adapting this framework on Sanskrit.

2. POS Tagging in Sanskrit

While POS tagging is not a new research topic, it is, indeed, a new field as far as Sanskrit is concerned. The usefulness of annotated corpora for natural language analyses is well known. Unfortunately, so far, there have been no such annotated corpora available for Sanskrit. An even greater problem has been the lack of training and testing data. The POS information is very important for language processing as it gives significant information about the word and its environment. This is not only true for major grammatical categories (i.e. verb *paṭhati* versus participle *paṭhati*: the verb *paṭhati* expects a noun in nominative but a participle *paṭhati* expects a noun in locative), but also for many other finer distinctions. For example, the declension sub-tags (attributes) and gender-number sub-tags can help distinguish words in different categories (as in *naraḥ gacchanti* the number tag in the verb *gacchanti* demands *naraḥ* to be the nominative plural of the base *nṛ*, and not the nominative plural of the word *nara*). By knowing the POS of a word we can tell which word with which POS label is likely to occur in its vicinity. In linguistic items, such complexity is found across languages.

POS ambiguities in Sanskrit can be enormous. Chandrashekar (2007) has found eight types of ambiguity in the context of POS tag application. In Sanskrit, the nominal base (*prātipadika*) is inflected for multiple information based on the end-character, gender (*liṅga*), *vibhakti* and number (*vacana*) information. The nominal forms (*subanta*-

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

rūpas) having *vibhakti* markings *-bhyām*, *-bhyas*, *-os* are ambiguous in all endings and genders of a nominal bases (*prātipadika*). Sometimes different nominal bases have similar forms (for e.g., *vibhavaḥ* [1.1] when the *prātipadika* is *vibhava* / *vibhavaḥ*[1.3] when the *prātipadika* is *vibhu*). (Chandrashekar 2007). In a verb, the verb root, along with the optional prefix information like *pada* (*ātmane/ parasmai*), transitivity (*karmatva*), tense (*kāla*), mode (*artha*), voice (*vācya*), person (*puruṣa*) and number (*vacana*) are clubbed together. Adding to the complication, the nominal and verbal bases may be derived bases. Such being the complexity of Sanskrit morphology, there is ample scope for ambiguous word forms.

The complexity of ambiguity in Sanskrit can be demonstrated using an example *bhavati*. It can be a verb or pronoun or a participle. It can still have many more ambiguous forms within the above mentioned categories, if we take inflectional features into consideration as well. Some forms of the first person pronoun *bhavat* in three genders can have similar forms as that of forms in the present participle of the verb root *bhu*. Though actual usage of the *bhavati* in the sense of participle and pronoun in neuter gender is often not seen, but grammaticality of the usage cannot be ruled out.

3. Sanskrit Morphology

In Sanskrit, a syntactic unit is called *pada*. Cordona (1988) posits the formula for Sanskrit sentence (N-En)p…(V-Ev)p. A *pada* can be nominal (*subanta*) or verbal (*tiṅanta*). *Padas* with *sup* (nominal) inflections constitute the NPs (*subanta-pada*), and those with *tiṅ* (verbal) can be called constituting the VPs (*tiṅanta-pada*). In the former, the bases are called *prātipadikas* which undergo sup affixations under specifically formulated conditions of case, gender, number, and also the end-characters of the bases to yield nominal syntactic words. The rules for *subanta padas* are found scattered in AD mostly in chapters 7-1, 7-2, 7-3, 6-1, 6-4. However, these rules have been treated in the *subanta* chapter of *Siddhānta Kaumudī* from rule number 177 to 446. (Jha 2004 b)

The derivational morphology in Sanskrit studies primary forms (*kṛdanta*) and secondary forms (*taddhita*), compounds (*samāsa*), feminine forms (*strī pratyaya*) etc (Subash 2006). These can be inflected for 21 case (7 cases x 3 number) affixes to generate 21 inflected forms.

The verb morphology (*tiṅanta*) is equally complex. Sanskrit has approximately 2014 verb roots including *kaṇḍvādi* according to Pāṇinian *dhātupāṭha* classified in 10 *gaṇas* to undergo peculiar operations (Jha 2004 a), it can also be sub-classified in 12 derivational suffixes. A verb root conjugates for tense, mood, number and person information. Further, these can have *ātmanepadī* and *parasmaipadī* forms in 10 *lakāras* and 3x3 person and number combinations. There are 12 secondary suffixes added to verb roots to create new verb roots. A verb root may have approximately 2190 (tense, aspect, number etc.) morphological forms. Mishra and Jha (2005) have done a rough calculation of all potential verb forms in Sanskrit to be more than 1029,60,000.

4. MSRI hierarchical tagset schema

MSRI in collaboration with linguists and NLP experts has developed a common POS Tagset framework for Indian languages (especially for Indo-Aryan and Dravidian languages) following the hierarchical and decomposable tagset schema similar to that of EAGLES. This framework facilitates the sharing and reusability of scarce resources in Indian languages and ensures cross-linguistic compatibility. The rationale behind concentrating on Dravidian and IA language families has been that of the 22 official languages in India a large majority is associated with these two language families.

The Dravidian and IA language families have very different morpho-syntactic features at every level of linguistic analyses, but they also have a number of typological similarities that facilitate a common framework.

Unlike flat tagsets, a hierarchical tagset exploits the linguistic hierarchy among categories. This implies that instead of having a large number of independent categories, a hierarchical tagset accommodates a small number of categories at the top level, each of which has a number of sub-categories in a tree structure. The associated morpho-syntactic features are packed in the different layers of hierarchy beginning from the major categories in the top and gradually progressing down to cover morpho-syntactic features for making it suitable to any Indian language, thereby keeping the framework a common standard across languages.

The hierarchical tagset requires another feature called 'decomposability'. It allows different features to be incorporated in a tag by separate sub-strings. Decomposable tags help in better corpus analysis (Leech, 1997) by allowing to search with an underspecified search string.

4.1. The IL-POSTS

This framework has a hierarchy at three levels:.

A. *Categories* are the highest level lexical classes. All categories are obligatory, i.e., are generally universal for all languages.

B. *Types* are subclasses of categories and are assumed to be significant sub-classes common to a majority of languages. Some types may also be optional for certain languages.

C. *Attributes* are morpho-syntactic features of types. Attribute tags contain the information like gender

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

(15)

(masculine, feminine, neuter), number (singular, dual, and plural), case (nominative, accusative etc.), person (first, second, third) etc. All attributes are optional, though in some cases they may be recommended.

The framework consists of 11 categories (including the punctuations and residual categories) that are recognised as universal categories for all ILs and hence, these are obligatory for any tagset derived from IL-POSTS.

Barring punctuations, all categories have sub-classes called types which can have a number of attributes belonging to each of them. There are 17 attributes defined in the IL-POSTS framework. The attributes can be either binary or multi-valued.

5. Adaptations for Sanskrit

For finalising our tagset, we have used the Sanskrit tagset by Chandrashekar (2007) and the Hindi specific tagset of MSRI. There are some changes in the tagset that we have adapted for Sanskrit data. These are at the category, subcategory, and attribute levels. In the subcategories of noun, we have only common and proper nouns. Verbal nouns have been clubbed under common nouns and spatio-temporal nouns under adverbs. Hindi and Sanskrit nouns do not have similar attributes. The verb has only finite form in Sanskrit and there is no auxiliary. The *upapada 'sma'* behaves like an auxiliary but we have put it under particle (*avyaya*). There is no change in the subtypes of pronoun. However, they differ in their attributes. The pronominal takes gender, number, person, case, nominal declension, emphatic, honorificity, and distance attributes. The reflexive and reciprocal carry gender, number, case, and nominal declension attributes. The relative pronoun has gender, number, person, case, and nominal declension. And finally, the Wh-pronoun takes gender, number, person, case, and nominal declension. The subtypes of the demonstrative are same as in Hindi. The adverb category in which a number of Sanskrit indeclinables fall has no attribute like Hindi adverbs. The postposition is not found in Sanskrit. Their role has been replaced by different declensions. In Sanskrit, we have participles (*kṛdanta*) with subtypes as participle proper and participle gerundive. They are marked for gender, number, case, and nominal declension attributes. Under the particle category we have two extra subtypes- negative and emphatic. Particles have no attribute. Punctuation and residual have the same conditions in our tagset.

6. Proposed IL-POSTS for Sanskrit

Following is the tagset we propose for annotating Sanskrit corpus. Using this tagset, we have annotated a corpus of simple Sanskrit text including articles and stories which are online available in our website at http://sanskrit.jnu.ac.in/corpora/annotated/MSRIndic-JNUTagsetTaggedCorpora.txt. Currently, annotation of several Sanskrit story-collections is in progress.

Category	Type	Attributes
Noun (N)	Common (NC)	gender, number, case, nominal declension
	Proper (NP)	gender, number, case, nominal declension
Verb (V)		pada, number, person, tense\mood, honorificity
Pronoun (P)	Pronominal (PPR)	gender, number, person, case, nominal declension, emphatic, honorificity, distance
	Reflexive (PRF)	gender, number, case, nominal declension
	Reciprocal (PRC)	gender, number, case, nominal declension
	Relative (PRL)	gender, number, person, case, nominal declension
	Wh (PWH)	gender, number, person, case, nominal declension
Nominal Modifier (J)	Adjective (JJ)	gender, number, case, nominal declension, emphatic, negative, honorificity
	Quantifier (JQ)	gender, number, case, nominal declension, numeral, emphatic, negative
Demonstrative (D)	Absolutive (DAB)	gender, number, person, case, nominal declension, distance, honorificity
	Relative (DRL)	gender, number, person, case, nominal declension, distance, honorificity
	Wh- (DWH)	gender, number, person, case, nominal declension, distance, honorificity
Kṛdanta (KD)	Participle (KDP)	gender, number, case, nominal declension
	Gerundive	gender, number, case, nominal

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

	(KDG)	declension
Particle (C)	Coordinating (CCD)	
	Subordinating (CSB)	
	Gerundive (CGD)	
	Interjection (CIN)	
	Negative (CNG)	
	Emphatic (CEM)	
	Interrogative (CNT)	
	Adverb (CAD)	
	Postposition (upapada) (CPP)	
	Quotative (CQT)	
	Comparative (CCM)	
	Reduplicative (CRD)	
	Other (CX)	
Punctuation (PU)		
Residual (RD)	Foreign word (RDF)	
	Symbol (RDS)	
	Others (RDX)	

Table1. Categories, types and their respective attributes for Sanskrit

No.	Attributes	Values
1.	Gender (Gen)	Masculine (mas), Feminine (fem), Neuter (neu)
2.	Number (Num)	Singular (sg),Dual (du), Plural (pl)
3.	Person (Per)	First (1), Second (2), Third (3)
4.	Case (Cs)	Nominative (nom), Accusative (acc), Instrumental (ins), Dative (dat), Ablative (abl), Genetive (gen), Locative (loc), vocative (voc),
5.	Nominal declension vibhakti (Vbh)	prathamā (i), dvitīyā (ii), tritīyā (iii), caturthī (iv), pañcamī (v), ṣaṣṭhī (vi), saptamī (vii), vocative (viii)
6.	Tense/Mood (Tns/Mood)	Present (prs), Aorist (aor), Imperfect (imprf), Perfect (prf), Periphrastic Future (phf), General Future (gft), Imperative (imp), Potential (pot), Benedictive (ben), Conditional (cnd
7.	Numeral (Nml)	Ordinal (ord), Cardinal (crd), Non-numeral (nnm)
8.	Distance (Dist)	Proximal (prx), Distal (dst)
9.	Emphatic (Emph)	Yes y, No n
10.	Negative (Neg)	Yes y, No n
11.	Honorificity (Hon)	Yes y, No n
12.	Pada (Pd)	parasmaipada (ppd), ātmanepada (apd)

Table2. Attributes and their values for Sanskrit

We have also used the following common attributes that MSRI Tagset contains:

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

17

- Not-applicable (0); when any other value is not applicable to the category or the relevant morpho-syntactic feature is not available.
- Undecided or doubtful (x); when the annotator is not sure about the exact attribute.

7. POS results and current status

The initial experiment of tagging 10 K data of ordinary Sanskrit and subsequent training of the tagger at MSRI is documented below -

Training data: 200 sentences (~5.8K words)
Test Data: 50 Sentences (~1.2K words)

Word Level Acc: 75.35%
Sentence Level Acc: 29.3%

In the similar experimental set up, the accuracy of Sanskrit was much better than Bangla (71%) and lower to Hindi (77%). Currently, we are tagging about 50 K data from two story collections - *Pañcatantra* and *Hitopadeśa*. This task will finish in about two months time. The subsequent training and automatic tagging results will be reported.

8. Conclusion

In this paper we have presented a Sanskrit specific tagset framework for annotating Sanskrit corpus. At experiment level a certain amount of data has been manually tagged and we have revised our tagset again and again. This framework follows the guidelines of the IL-POSTS framework for Indic Languages as much as possible. We have tried our level best to be near to this hierarchical framework. It has also been observed that IL-POSTS framework is adaptable for Sanskrit as well. This Sanskrit Tagset along with the annotation guidelines (that we ourselves have designed for tagging Sanskrit text) and tagged corpus is available on our website: http://sanskrit.jnu.ac.in.

References

AU-KBC tagset. AU-KBC POS tagset for Tamil. Retrieved from http://nrcfosshelpline.in/smedia/images/downloads/Tamil_Tagset-opensource.odt

Baskaran, S. Kalika Bali, Tanmoy Bhattacharya, Pushpak Bhattacharyya, Monojit Choudhury, Girish Nath Jha, Rajendran S., Saravanan K., Sobha L., KVS Subbarao (2008). *A Common Parts-of-Speech Tagset Framework for Indian Languages,* LREC 2008 - 6th Language Resources and Evaluation Conference, May 26-June1, 2008, Marrakech, Morocco

Baskaran S. Kalika Bali, Tanmoy Bhattacharya, Pushpak Bhattacharyya, Monojit Choudhury, Girish Nath Jha, Rajendran S., Saravanan K., Sobha L., KVS Subbarao (2008). *Designing a Common POS-Tagset Framework for Indian Languages.* The 6thWorkshop on Asian Language Resources. January, 2008, Hyderabad.

Cardona, George, 1988. *Pāṇini: His work and its traditions*, Motilal Banarasidass, Delhi

Chandrashekar R., 2007. *POS Tagger for Sanskrit*, Ph.D. thesis, Jawaharlal Nehru University

Cloeren, J., 1999. *Tagsets. In Syntactic Wordclass Tagging*, ed. Hans van Halteren, Dordrecht.: Kluwer Academic

Jha Girish Nath, 2004. *Generating nominal inflectional morphology in Sanskrit*, SIMPLE 04, IIT-Kharagpur Lecture Compendium, Shyama Printing Works, Kharagpur, WB

Jha Girish Nath, Sobha L, Diwakar Mishra, Surjit K Singh, Praveen Pralayankar, 2007. *Sanskrit Anaphors*, Johansson, C. (Ed.) Proceedings of the Second Workshop on Anaphora Resolution (2008), ISSN 1736-6305 Vol. 2, Cambridge Scholars Publishing, 2007

Jha Girish Nath, Mishra Sudhir, 2009. *Semantic processing in Panini's karaka system*, Lecture Notes in Computer Science series 5402 Springer Berlin / Heidelberg, Gérard P. Huet, Amba P. Kulkarni, Peter Scharf (eds)

Greene, B.B. and Rubin, G.M.,1981. *Automatic grammatical tagging of English*. Providence, R.I.: Department of Linguistics, Brown University.

Hardie, A., 2004. *The Computational Analysis of Morphosyntactic Categories in Urdu.* PhD Thesis submitted to Lancaster University.

IIIT-Tagset. *A Parts-of-Speech tagset for Indian Languages.* Retrieved from http://shiva.iiit.ac.in/SPSAL2007/iiit_tagset_guidelines.pdf

Huet Gerard, *The Sanskrit Heritage Site*, http://sanskrit.inria.fr/

Kale, M.R., 1995. *A Higher Sanskrit Grammar*, MLBD Publishers, New Delhi

Leech, G and Wilson, A., 1996. *Recommendations for the Morphosyntactic Annotation of Corpora.* EAGLES Report EAG-TCWG-MAC/R.

Leech, G and Wilson, A., 1999. *Standards for Tag-sets.* In Syntactic Wordclass Tagging, ed. Hans van Halteren, Dordrecht: Kluwer Academic.

Leech, G., 1997. *Grammatical Tagging.* In Corpus Annotation: Linguistic Information for Computer Text Corpora, ed: Garsire, Leech, and McEnery, London: Longman.

Mishra Sudhir, Jha, Girish Nath, 2005. *Identifying verb inflections in Sanskrit morphology*, proceedings of SIMPLE'04, IIT Kharagpur

NLPAI Contest- 2006, retrieved from http://ltrs.iiit.ac.in/nlpai_cntest06

Santorini, B., 1990. *Part-of-speech tagging guidelines for the Penn Treebank Project.* Technical report MS-CIS-90-47, Dept. Of Computer and Information Science, University of Pennsylvania.

Subash, 2006. *Sanskrit Subanta Recognizer and Analyzer*, M.Phil dissertation submitted to Jawaharlal Nehru University, New Delhi

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

ACL Digitized by Sarayu Trust Foundation Delhi and eGangotri.

Automatic Sanskrit Segmentizer Using Finite State Transducers

Vipul Mittal
Language Technologies Research Center, IIIT-H,
Gachibowli, Hyderabad, India.
vipulmittal@research.iiit.ac.in

Abstract

In this paper, we propose a novel method for automatic segmentation of a Sanskrit string into different words. The input for our segmentizer is a Sanskrit string either encoded as a Unicode string or as a Roman transliterated string and the output is a set of possible splits with saliency associated with each of them. We followed two different approaches to segment a Sanskrit text using *sandhi*[1] rules extracted from a parallel corpus of manually sandhi split text. While the first approach augments the finite state transducer used to analyze Sanskrit morphology and traverse it to segment a word, the second approach generates all possible segmentations and validates each constituent using a morph analyzer.

1 Introduction

Sanskrit has a rich tradition of oral transmission of texts and this process causes the text to undergo euphonic changes at the word boundaries. In oral transmission, the text is predominantly spoken as a continuous speech. However, continuous speech makes the text ambiguous. To overcome this problem, there is also a tradition of reciting the pada-pāṭha (recitation of words) in addition to the recitation of a saṃhitā (a continuous sandhied text). In the written form, because of the dominance of oral transmission, the text is written as a continuous string of letters rather than a sequence of words. Thus, the Sanskrit texts consist of a very long sequence of phonemes, with the word boundaries having undergone euphonic changes. This makes it difficult to split a continuous string into words and process the text automatically.

Sanskrit words are mostly analyzed by building a finite state transducer. In the first approach, this transducer was modified by linking the final states to appropriate intermediate states incorporating the sandhi rules. This approach then allows one to traverse the string from left to right and generate all and only possible splits that are morphologically valid. The second approach is very closely based on the *Optimality Theory* (Prince and Smolensky, 1993) where we generate all the possible splits for a word and validate each using a morphological analyzer. We use one of the fastest morphological analyzers available viz. the one developed by *Apertium* group[2]. The splits that are not validated are pruned out. Based on the number of times the first answer is correct, we achieved an accuracy of around 92% using the second approach while the first approach performed with around 71% accuracy.

2 Issues involved in Sanskrit Processing

The segmentizer is an important component of an NLP system. Especially, languages such as Chinese (Badino, 2004), Japanese, Thai (Haruechaiyasak, 2008) or Vietnamese (Thang et al. , 2008) which do not mark word boundaries explicitly or highly agglutinative languages like Turkish need segmentizers. In all these languages, there are no explicit delimiters to specify the word boundaries. In Thai, each syllable is transcribed using several characters and there is no space in the text between syllables. So the problem of segmentation is basically twofold: (1) syllable segmentation followed by (2) word segmentation itself. A sentence in these languages

[1]*Sandhi* means euphony transformation of words when they are consecutively pronounced. Typically when a word w_1 is followed by a word w_2, some terminal segment of w_1 merges with some initial segment of w_2 to be replaced by a "smoothed" phonetic interpolation, corresponding to minimizing the energy necessary to reconfigurate the vocal organs at the juncture between the words.

[2]http://wiki.apertium.org/wiki/lttoolbox; It processes around 50,000 words per sec.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

is segmented by predicting the word boundaries, where euphonic changes do not occur across the word boundaries and it is more like mere concatenation of words. So the task here is just to choose between various combinations of the words in a sentence.

However, in Sanskrit, euphonic changes occur across word boundaries leading to addition and deletion of some original part of the combining words. These euphonic changes in Sanskrit introduce non-determinism in the segmentation. This makes the segmentation process in Sanskrit more complex than in Chinese or Japanese. In case of highly agglutinative languages like Turkish, the components are related to each other semantically involving dependency analysis. Whereas in Sanskrit, only the compounds involve a certain level of dependency analysis, while sandhi is just gluing of words together, without the need for words to be related semantically. For example, consider the following part of a verse,

San: *nāradam paripapraccha vālmīkirmunipuṅgavam*
gloss: to_the_Narada asked Valmiki-to_the_wisest_among_sages
Eng: Valmiki asked the Narada, the wisest among the sages.

In the above verse, the words *vālmīkiḥ* and *munipuṅgavam* (wisest among the sages - an adjective of Narada) are not related semantically, but still undergo euphonic change and are glued together as *vālmīkirmunipuṅgavam*.

Further, the split need not be unique. Here is an example, where a string *māturājñāmparipālaya* may be decomposed in two different ways after undergoing euphonic changes across word boundaries.

- *mātuh ājñām paripālaya* (obey the order of mother) and,
- *mā āturājñām paripālaya* (do not obey the order of the diseased).

There are special cases where the sandhied forms are not necessarily written together. In such cases, the white space that physically marks the boundary of the words, logically refers to a single sandhied form. Thus, the white space is deceptive, and if treated as a word boundary, the morphological analyzer fails to recognize the word. For example, consider

śrutvā ca nārado vacaḥ.

In this example, the space between *śrutvā* and *ca* represent a proper word boundary and the word *śrutvā* is recognized by the morphological analyzer whereas the space between *nārado* and *vacaḥ* does not mark the word boundary making it deceptive. Because of the word *vacaḥ*, *nāradaḥ* has undergone a phonetic change and is rendered as *nārado*. In unsandhied form, it would be written as,

San: *śrutvā ca nāradaḥ vacaḥ.*
gloss: after_listening and Narada's speech
Eng: And after listening to Narada's speech

The third factor aggravating Sanskrit segmentation is productive compound formation. Unlike English, where either the components of a compound are written as distinct words or are separated by a hyphen, the components of compounds in Sanskrit are always written together. Moreover, before these components are joined, they undergo the euphonic changes. The components of a compound typically do not carry inflection or in other words they are the bound morphemes used only in compounds. This forces a need of a special module to recognize compounds.

Assuming that a sandhi handler to handle the sandhi involving spaces is available and a bound morpheme recognizer is available, we discuss the development of sandhi splitter or a segmentizer that splits a continuous string of letters into meaningful words. To illustrate this point, we give an example.

Consider the text,

śrutvā caitattrilokajño vālmīkernārado vacaḥ.

We assume that the sandhi handler handling the sandhi involving spaces is available and it splits the above string as,

śrutvā caitattrilokajñaḥ vālmīkernāradaḥ vacaḥ.

The sandhi splitter or segmentizer is supposed to split this into

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

śrutvā ca etat triloka-jñaḥ vālmīkeḥ nāradaḥ vacaḥ.

This presupposes the availability of rules corresponding to euphonic changes and a good coverage morphological analyzer that can also analyze the bound morphemes in compounds.

A segmentizer for Sanskrit developed by Huet (Huet, 2009), decorates the final states of its finite state transducer handling Sanskrit morphology with the possible sandhi rules. However, it is still not clear how one can prioritize various splits with this approach. Further, this system in current state demands some more work before the sandhi splitter of this system can be used as a standalone system allowing plugging in of different morphological analyzers. With a variety of morphological analyzers being developed by various researchers[3], at times with complementary abilities, it would be worth to experiment with various morphological analyzers for splitting a sandhied text. Hence, we thought of exploring other alternatives and present two approaches, both of which assume the existence of a good coverage morphological analyzer. Before we describe our approaches, we describe the baseline system and also define the saliency which will be used to prioritize various analyses.

3 Saliency

Just as in the case of any NLP systems, with the sandhi splitter being no exception, it is always desirable to produce the most likely output when a machine produces multiple outputs. To ensure that the correct output is not deeply buried down the pile of incorrect answers, it is natural to prioritize solutions based on some frequencies. A Parallel corpus of Sanskrit text in sandhied and sandhi split form is being developed as a part of the Consortium project in India. Around 600K words of such a parallel corpus is available. The corpus contains texts from various fields ranging from children stories, dramas, to Ayurvedas. We have used around 25,000 extracted parallel strings of unsandhied and corresponding sandhied texts to extract a total of 2650 sandhi rules including the cases of mere concatenation. Each sandhi rule is a triple (x, y, z) where y is the last letter of the first primitive, z is the first letter of the second primitive, and x is the letter sequence created by euphonic combination. The same corpus was used to calculate the frequencies of occurrence of these sandhi rules. We define the estimated probability of the occurrence of a sandhi rule as follows:

Let R_i denote the i^{th} rule with f_{R_i} as the frequency of occurrence in the manually split parallel text. The probability of rule R_i is:

$$P_{R_i} = \frac{f_{R_i}}{\sum_{i=1}^{n} f_{R_i}}$$

where n denotes the total number of sandhi rules found in the corpus.

Let a word be split into a candidate S_j with k constituents as $< c_1, c_2, ..c_k >$ by applying $k-1$ sandhi rules $< R_1, R_2, ..R_{k-1} >$ in between the constituents. It should be noted here that the rules $R_1, ..R_{k-1}$ and the constituents $c_1, ..c_k$ are interdependent since a different rule sequence will result in a different constituents sequence. Also, except c_1 and c_k, all intermediate constituents take part in two segmentations, one as the right word and one as the left.

The saliency of the split S_j is defined as:

$$W_{S_j} = \frac{\prod_{x=1}^{k-1}(P_{c_x} + P_{c_{x+1}}) * P_{R_x}}{k}$$

where P_{c_x} is the probability of occurrence of the word c_x in the corpus. The factor of k was introduced to give more preference to the split with less number of segments than the one with more segments.

4 Baseline System

We define our own baseline system which assumes that each Sanskrit word can be segmented only in two constituents. A word is traversed from left to right and is segmented by applying the first applicable rule provided both the constituents are valid morphs. Using the 2,650 rules, on a test data of 2,510 words parallel corpus, the baseline performance of the system was around 52.7% where the first answer was correct.

5 Two Approaches

We now present the two approaches we explored for sandhi splitting.

5.1 Augmenting FST with Sandhi rules

In this approach, we build an FST, using OpenFst (Allauzen et al., 2007) toolkit, incorporating

[3]http://sanskrit.uohyd.ernet.in, http://www.sanskritlibrary.org, http://sanskrit.jnu.ernet.in

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

sandhi rules in the FST itself and traverse it to find the sandhi splittings.

We illustrate the augmentation of a sandhi rule with an example. Let the two strings be *xaXi* (dadhi)[4] and *awra* (atra). The initial FST without considering any sandhi rules is shown in Figure 1.

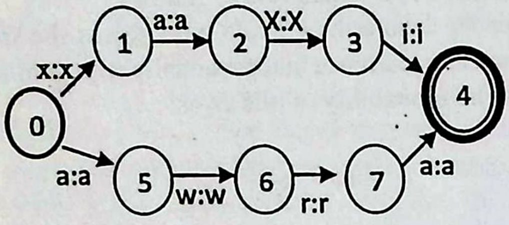


Figure 1: Initial FST

As the figure depicts, 0 is the start state and 4 is the final state. Each transition is a 4-tuple <*c, n, i, o*> where *c* is current state, *n* is the next state, *i* is the input symbol and *o* is the output. The FST marks word boundaries by flushing out certain features about the words whenever it encounters a valid word. Multiple features are separated by a |. E.g., the output for *xaXi* is *lc,s|vc,s* and for *awra* it is *vc,s* where *lc,s* stands for *locative, singular* and *vc,s* is *vocative, singular*. The FST in Figure 1 recognize exactly two words *xaXi* and *awra*.

One of the sandhi rule states that $i{+}a \rightarrow ya$ which will be represented as a triple (ya, i, a). Applying the sandhi rule, we get: *xaXi* + *awra* $\rightarrow$ *xaXyawra*. After adding this sandhi rule to the FST, we get the modified FST that is represented in Figure 2.

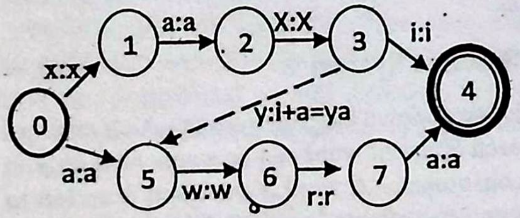


Figure 2: Modified FST after inserting the rule. – – – indicates the newly added transition.

Here, a transition arc is added depicting the rule which says that on receiving an input symbol *ya* at state 3, go to state 5 with an output $i{+}a \rightarrow ya$. Thus the new FST accepts *xaXyawra* in addition to *xaXi* and *awra*.

[4]A Roman transliteration scheme called WX transliteration is used, which is one-to-one phoneme level representation of Devanāgarī script.

Thus, we see that the original transducer gets modified with all possible transitions at the end of a final phoneme, and hence, also explodes the number of transitions leading to a complex transducer.

The basic outline of the algorithm to split the given string into sub-strings is:

Algorithm 1 To split a string into sub-strings

1: Let the FST for morphology be *f*.
2: Add sandhi rules to the final states of *f1* linking them to the intermediary states to get f'.
3: Traverse f' to find all possible splits for a word. If a sandhi rule is encountered, split the word and continue with the remaining part.
4: Calculate the saliency of the possible outputs with the formula discussed in section 3.

The pseudo-code of the algorithm used to insert sandhi rules in the FST is illustrated here:

Algorithm 2 To insert sandhi rules in the FST

```
1: I = Input Symbol; X = last character of the
   result of the rule.
2: for each transition in the FST transition table
   do
3:   if next state is a final state then
4:     for all rules where I is the last character
       of first word do
5:       S = next state from the start state on
         encountering X;
6:       Y = first character of the result of the
         rule;
7:       transition T = current state, S, Y, rule;
8:       Add T into the FST;
9:     end for
10:  end if
11: end for
```

The main problem with this approach is that every finite state can have as many transitions as the number of euphonic rules resulting in phoneme change. This increases the size of the FST considerably. It should be noted that, we have not included the cases, where there is just a concatenation. In such cases, if the input string is not exhausted, but the current state is a final state, we go back to the start state with the remaining string as the input.

5.1.1 Results

The performance of this system measured in terms of the number of times the highest ranked segmen-

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

tation is correct, with around 500 sandhi rules, and only noun morphology tested on the same test data used for testing baseline system gave the following rank-wise distribution presented in Table 1.

Rank	% of output
1	71.2509
2	5.64543
3	3.85324
4	3.35651
5	1.56123
>5	14.33268

Table 1: Rank-wise Distribution for Approach-1.

The system was slow consuming, on an average, around 10 seconds per string of 15 letters.[5].

With the increase in the sandhi rules, though system's performance was better, it slowed down the system further. Moreover, this was tested only with the inflection morphology of nouns. The verb inflection morphology and the derivational morphology were not used at all. Since, the system is supposed to be part of a real time application viz. machine translation, we decided to explore other possibilities.

5.2 Approach based on Optimality Theory

Our second approach follows optimality theory(OT) which proposes that the observed forms of a language are a result of the interaction between the conflicting constraints. The three basic components of the theory are:

1. GEN - generates all possible outputs, or candidates.

2. CON - provides the criteria and the constraints that will be used to decide between candidates.

3. EVAL - chooses the optimal candidate based on the conflicts on the constraints.

OT assumes that these components are universal and the grammars differ in the way they rank the universal constraint set, CON. The grammar of each language ranks the constraints in some dominance order in such a way that every constraint must have outperformed every lower ranked constraint. Thus a candidate A is optimal if it performs better than some other candidate B on a higher ranking constraint even if A has more violations of a lower ranked constraint than B.

[5]Tested on a system with 2.93GHz Core 2 Duo processor and 2GB RAM

The GEN function produces every possible segmentation by applying the rules wherever applicable. The rules tokenize the input surface form into individual constituents. This might contain some insignificant words that will be eventually pruned out using the morphological analyser in the EVAL function thus leaving the winning candidate. Therefore, the approach followed is very closely based on optimality theory. The morph analyser has no role in the generation of the candidates but only during their validation thus composing the back-end of the segmentizer. In original OT, the winning candidate need not satisfy all the constraints but it must outperform all the other candidates on some higher ranked constraint. While in our scenario, the winning candidate must satisfy all the constraints and therefore there could be more than one winning candidates.

Currently we are applying only two constraints. We are planning to introduce some more constraints. The constraints applied are:

- C1 : All the constituents of a split must be valid morphs.

- C2 : Select the split with maximum saliency, as defined in section 3.

The basic outline of the algorithm is:

1: Recursively break a word at every possible position applying a sandhi rule and generate all possible candidates for the input.
2: Pass the constituents of all the candidates through the morph analyzer.
3: Declare the candidate as a valid candidate, if all its constituents are recognized by the morphological analyzer.
4: Assign weights to the accepted candidates and sort them based on the saliency as defined in section 4.
5: The optimal solution will be the one with the highest salience.

5.2.1 Results

The current morphological analyzer can recognize around 140 million words. Using the 2650 rules and the same test data used for previous approach, we obtained the following results:

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

- Almost 93% of the times, the highest ranked segmentation is correct. And in almost 98% of the cases, the correct split was among the top 3 possible splits.

- The system consumes around 0.04 seconds per string of 15 letters on an average.

The complete rank wise distribution is given in Table 2.

	% of output	
Rank	Approach-1	Approach-2
1	71.2509	92.8771
2	5.64543	5.44693
3	3.85324	1.07076
4	3.35651	0.41899
5	1.56123	0.09311
>5	14.33268	0.0931

Table 2: Complete rank-wise Distribution.

6 Conclusion

We presented two methods to automatically segment a Sanskrit word into its morphologically valid constituents. Though both the approaches outperformed the baseline system, the approach that is close to optimality theory gives better results both in terms of time consumption and segmentations. The results are encouraging. But the real test of this system will be when it is integrated with some real application such as a machine translation system. This sandhi splitter being modular, wherein one can plug in different morphological analyzer and different set of sandhi rules, the splitter can also be used for segmentization of other languages.

In future work, the major task would be to explore ways to shift rank 2 and rank 3 segmentations more towards rank 1. We are also exploring the possibility of including some semantic information about the words while defining saliency. The sandhi with white spaces also needs to be handled.

Acknowledgments

I would like to express my gratitude to Dr. Amba P. Kulkarni and Prof. Rajeev Sangal for their guidance and support.

References

Akshar Bharati, Amba P. Kulkarni, and V Sheeba. 2006. *Building a wide coverage Sanskrit morphological analyzer: A practical approach.* The First National Symposium on Modelling and Shallow Parsing of Indian Languages, IIT-Bombay.

Alan Prince and Paul Smolensky. 1993. *Optimality Theory: Constraint Interaction in Generative Grammar.* RuCCS Technical Report 2 at Center for Cognitive Science, Rutgers University, Piscataway.

Amba Kulkarni and Devanand Shukla. 2009. *Sanskrit Morphological analyzer: Some Issues.* To appear in Bh.K Festschrift volume by LSI.

Choochart Haruechaiyasak, Sarawoot Kongyoung, and Matthew N. Dailey. 2008. *A Comparative Study on Thai Word Segmentation Approaches.* ECTI-CON, Krabi.

Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. *OpenFst: A General and Efficient Weighted Finite-State Transducer Library.* CIAA'07, Prague, Czech Republic.

Deniz Yuret and Ergun Biçici. 2009. *Modeling Morphologically Rich Languages Using Split Words and Unstructured Dependencies.* ACL-IJCNLP'09, Singapore.

DINH Q. Thang, LE H. Phuong, NGUYEN T. M. Huyen, NGUYEN C. Tu, Mathias Rossignol, and VU X. Luong. 2008. *Word Segmentation of Vietnamese Texts: a Comparison of Approaches.* LREC'08, Marrakech, Morocco.

Gérard Huet. 2009. *Formal structure of Sanskrit text: Requirements analysis for a mechanical Sanskrit processor.* Sanskrit Computational Linguistics 1 & 2, pages 266-277, Springer-Verlag LNAI 5402.

John C. J. Hoeks and Petra Hendriks. 2005. *Optimality Theory and Human Sentence Processing: The Case of Coordination.* Proceedings of the 27th Annual Meeting of the Cognitive Science Society, Erlbaum, Mahwah, NJ, pp. 959–964.

Leonardo Badino. 2004. *Chinese Text Word-Segmentation Considering Semantic Links among Sentences.* INTERSPEECH 2004 - ICSLP , Jeju, Korea.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Sanskrit Computational Linguistics Symposium 2010 Dec
Digitized by Sarayu Trust Foundation Delhi and eGangotri.

Designing a Constraint Based Parser for Sanskrit

Amba Kulkarni, Sheetal Pokar, and Devanand Shukl

Department of Sanskrit Studies,
University of Hyderabad,
Hyderabad
apksh@uohyd.ernet.in,{sjpokar,dev.shukl}@gmail.com

Abstract. Verbal understanding (śābdabodha) of any utterance requires the knowledge of how words in that utterance are related to each other. Such knowledge is usually available in the form of cognition of grammatical relations. Generative grammars describe how a language codes these relations. Thus the knowledge of what information various grammatical relations convey is available from the generation point of view and not the analysis point of view. In order to develop a parser based on any grammar one should then know precisely the semantic content of the grammatical relations expressed in a language string, the clues for extracting these relations and finally whether these relations are expressed explicitly or implicitly. Based on the design principles that emerge from this knowledge, we model the parser as finding a directed Tree, given a graph with nodes representing the words and edges representing the possible relations between them. Further, we also use the Mīmāṁsā constraint of ākāṅkṣā (expectancy) to rule out non-solutions and sannidhi (proximity) to prioritize the solutions. We have implemented a parser based on these principles and its performance was found to be satisfactory giving us a confidence to extend its functionality to handle the complex sentences.[1]

Key Words: Sanskrit, Constraint Based Parser, Information coding, *ākāṅkṣā, sannidhi.*

1 Introduction

Śābdabodha is the understanding that arises from a linguistic utterance. The three schools of Śābdabodha viz. *Vyākaraṇa*, *Nyāya* and *Mīmāṁsā* mainly differ in the chief qualificand of the Śābdabodha. Nevertheless to begin with, all these three schools need an analysis of an utterance. This analysis expresses the relations between different meaningful units involved in an utterance. The utterance may be as small as a single word or as big as a complete novel. In what follows, however, we take a sentence[2] as a unit, and as such we discuss

[1] Thanks to Gérard Huet and Peter Scharf for their valuable remarks.
[2] roughly *ekatiṅ vākyam (vārttika on tvāmau dvitīyāyāḥ 8.1.23, halantapuṁlliṅgaprakaraṇam).*

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

26

only the relations of words within a sentence and do not deal with the discourse analysis.

A generative grammar of any language provides rules for generation. For analysis, we require a mechanism by which we can use these rules in a reverse way. The reversal in some cases is easy and also deterministic. For example, subtraction is an inverse operation of addition and is deterministic. The reversal may not always be deterministic. Let us see a simple example of non-deterministic reversal with which all of us are familiar. The multiplication tables or simple method of repetitive addition provides a mechanical way for multiplication. Given a product, to find its factors is a reverse process. Multiplication of two numbers, say, 4 and 3 produces a unique number 12. But its decomposition into two factors is not unique. 12 may be decomposed into two factors as either {6,2} or {4,3} in addition to a trivial decomposition {12,1}. Thus the inverse process may at times involve non-determinism. Depending upon the context, if one factor is known, the other factor gets fixed. For example, if you are interested in distributing 12 apples among 2 children, then one of the factors being 2, the other factor, viz. 6, is determined uniquely.

This is true of a generative grammar as well. To give an example, look at the following two sūtras of Pāṇini.

– anabhihite (2.3.1)
– kartṛkaraṇayos tṛtīyā (2.3.18)

These two sūtras together, in case of a passive voice (karmaṇi prayogaḥ), assign third case[3] to both the kartā as well as karaṇam kāraka as in

(1) *rāmeṇa bāṇena vāliḥ hanyate.*

Now, when a hearer (who knows Sanskrit grammar) listens to this utterance, he notices two words ending in the third case suffix and that the construction is in passive voice. But unless he knows that *rāma* is the name of a person and *bāṇa* is used as an instrument, he fails to get the correct reading. In the absence of such an 'extra-linguistic' knowledge, there are two possible interpretations viz. either *rāma* is kartā and *bāṇa* is karaṇam, or *bāṇa* is kartā and *rāma* is karaṇam leading to a non-determinism.[4]

The process of analysing a sequence of words to determine the underlying grammatical structure with respect to a grammar is parsing. There are two distinct ways of developing a parser for a language. The first method which has gained recent predominance is to use statistical machine learning teshniques to learn from a manually annotated corpus. This requires a large human annotated

[3] the word 'case' is used for *vibhakti.*
[4] There are two more possibilities, since both have the same gender, number, and vibhakti, one can be an adjective of the other.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

corpus. Second method is to use the grammar rules of generation to 'guess' the possible solutions and apply constraints to rule out obvious non-solutions. There have been notable efforts in developing parsers by both the statistical methods as well as grammar based methods for various languages (Lin,1998; Marneffe, 2006; Sleator,1993). A parser based on Pāṇinian Grammar Formalism for modern Indian languages is described in Bharati, et. al. (1995; 85-100). This parser is modeled as a bipartite graph matching problem. A statistical parser for analysing Sanskrit is described in Hellwig (2008). The shallow parser of Huet (2006, 2009) uses bare minimum information of transitivity of a verb as a sub-categorisation frame and models it as a graph-matching algorithm. The main purpose of this shallow parser is to filter out non-sensical interpretations. It is therefore natural for Huet to develop small tools such as 'ca' handler with more priority to rule out non-grammatical solutions (rather than to develop a full-fledged parser)

While designing a grammar based parser, two major design issues[5] one has to address are: a) what should be the level of semantic analysis, and b) which relations to represent in the parsed output. In order to decide on these issues, in what follows, we first look at the Sanskrit grammar to see what kind of semantic relations can be extracted from a language string, precisely where is the information about these relations coded, and whether the extracted relations are from primary sources or secondary. Later we discuss the issues the mechanical processing throws up, and the possible ways to handle them. Based on these observations, we decide various design parameters. The next section discusses mathematical formulation of the problem, its implementation and finally its performance analysis.

2 Encoding of grammatical relations in Sanskrit

Parsing unfolds a linear string of words into a structure which shows explicitly the relations between words. For example, the parse of

(2) *rājā viprāya gāṁ dadāti.*

may be described as in Figure 1.

The task of a parser involves identifying various relations between the words. So the parser developer should decide on the nature of relations and the means to identify the relations. Sanskrit has the unique privilege of having an extant grammar in the form of Aṣṭādhyāyī. It has been demonstrated (Bharati, forthcoming) that Pāṇini had given utmost importance to the information coding and the dynamics of information flow in a language string. In what follows we

[5] The issues in the development of a statistical parser are totally different. They are related to the size of the annotated corpus, the number of annotated tags used, their fine-grained-ness, etc.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

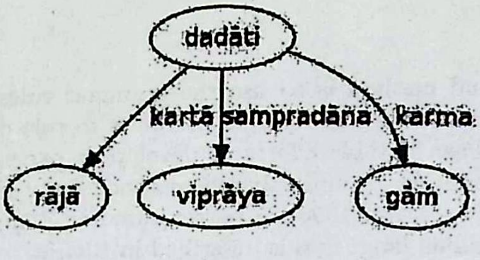


Fig. 1. semantic relations

look at the information coding in Sanskrit from the point of view of designing a parser.

2.1 Semantic content of the relations

Though the correspondence between the semantic relations and the kāraka relations is duly stated in the grammar, what is encoded in words is only the kāraka relations. There is no one-to-one relation between thematic and kāraka relations. One kāraka relation may correspond to more than one thematic relation and one thematic relation may be realised by more than one kāraka relations (Kiparsky, 2009: 49). What can be extracted from a language string alone without using any extra-linguistic information are the syntactico-semantic relations or the kāraka relations and not the pure semantic relations. We give below some examples in our support.

Svatantraḥ kartā The vārttikas under Pāṇini's sūtra *kārake* (1.4.23) go like this[6]

> In the sentence *devadattaḥ pacati*, the activity of cooking refers to the activity of devadattaḥ viz. putting a vessel on the stove, pouring water in it, adding rice, supplying the fuel etc. and this activity refers to the activity of the pradhāna kartā. In the sentence *sthālī pacati*, the cooking activity refers to holding the rice and water till the rice cooks and this activity is that of a vessel. In the sentence *edhāḥ pakṣyanti*, the cooking activity refers to the supply of sufficient heat by a piece of firewood and thus refers to the activity of an instrument.

[6] adhiśrayaṇodakāsecanataṇḍulāvapanaidho'pakarṣaṇakriyāḥ pradhānasya kartuḥ pākaḥ ||(ma. bhā. 1.4.23. vā 8) ||
droṇaṁ pacatyāḍhakaṁ pacatīti saṁbhavanakriyā dhāraṇakriyā cādhikaraṇasya pākaḥ ||(ma. bhā. 1.4.23.vā 9)||
edhāḥ pakṣyantyā viklitter jvaliṣyantīti jvalanakriyā karaṇasya pākaḥ ||(ma. bhā. 1.4.23.vā 10) ||

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri

In real world, *devadattaḥ*, *sthālī* and *edhāḥ* are the agent, locus and the instrument respectively. But what is expressed by these language strings is just the *kartṛtva* of the pradhāna kartā, adhikaraṇam and karaṇam respectively and NOT the agent, locus and instrument.

śeṣe Similarly the relation between *vṛkṣa* and *śākhā*, *pitṛ* and *putra*, and *rājan* and *puruṣa* in the phrases *vṛkṣasya śākhā*, *pituḥ putra.h* and *rājñaḥ puruṣaḥ* is marked by the genitive case suffix, and Pāṇini groups all of them under the sūtra *ṣaṣṭhī śeṣe* (2.3.50). Semantically however the first is avayava-avayavī-bhāva (part-whole-relation), the second one is janya-janaka-bhāva (parent-child-relation), and the third one is sva-svāmi-bhāva (owner-possession-relation).

***adhiśīṅsthāsāṁ karma* (1.4.46)** In the sentences

(3) ***hariḥ vaikuṇṭham adhiśete.***
(4) ***muniḥ śilāpaṭṭam adhitiṣṭhati.***
(5) ***sādhuḥ parvatam adhyāste.***

vaikuṇṭha, *śilāpaṭṭa* and *parvata* are in the second case, and Pāṇini assigns them a karma role. However, semantically, all of them are the loci of the activities of the associated verbs viz. adhi-śīṅ, adhi-sthā, and adhi-ās. Hence the naiyāyikas, who want to map the 'world of words' to the real world, find it difficult to accept the karmatva of these words and they qualify this karmatva on the second case ending as *ādhārasya anuśāsanika-karmatva* (Dash, 1991;141). Thus, there is a deviation between the real world and what is expressed through the words.

sahayukte 'apradhāne (2.3.19) In the sentence,

(6) ***mātrā saha bālakaḥ āgacchati.***

the agreement of the verb is with *bālakaḥ*, and not with *mātarā*. According to the sūtra (2.3.19), 'saha' is used with the apradhāna (sub-ordinate) kāraka. Thus in this example, *mātā* is sub-ordinate and *bālaka* is the main kartā. However, at another level of semantic analysis, the situation is reversed. It is *mātā* who carries the child in her arms and thus *bālaka* is apradhāna and *mātā* is the pradhāna kāraka. Thus again there is a mismatch between the reality and what sentence actually codes in terms of grammatical relations.

From all the above examples, it is clear that the world of words (śabda-jagat) is different from the real world. To match the extracted relations with the experience of the real world, extra-linguistic information is needed. Since the extra-linguistic information is not easily accessible, and is open ended, we would extract only syntactico-semantic relations that depend solely upon the linguistic

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

/ grammatical information in a sentence.

2.2 Clues for extracting the relations

Sanskrit being inflectionally rich, we know that suffixes mark the relation between words. Similarly certain indeclinables mark some grammatical relations. Agreement between the words also indicate certain grammatical relations. We discuss below these cases with examples.

1. Abhihitatva
   The Pāṇinian sūtra 'anabhihite' (2.3.1) (if not already expressed) is an important sūtra that governs the vibhakti assignment to the nominals. The vārttika[7] on this sūtra explains abhihita as the one which is expressed either by tiṅ (a finite verbal suffix), kṛt (a non-finite verbal suffix), taddhita (derivational nominal suffix) or samāsa (compound). E.g. in the sentence

   (7) *rāmaḥ vanaṁ gacchati.*

   the verb being in the active voice (kartari prayogaḥ), the verbal suffix 'ti' expresses the kartā, while in the following sentence in passive voice (karmaṇi prayogaḥ)

   (8) *rāmeṇa vanaṁ gamyate.*

   the karma is expressed by the verbal suffix. As such, in both cases, the one which is expressed (kartā and karma respectively) is in the nominative case and shows number and person agreement with the verb form.

   Some of the kṛt suffixes also express the kārakas. For example, in

   (9) *dhāvan aśvaḥ.*

   the kṛt suffix in 'dhāvan' expresses the relation of kartā (kartari kṛt (3.4.68)).

2. Vibhakti
   The verbal as well as nominal suffixes in Sanskrit are termed vibhaktis. We have already seen that verbal suffixes (*tiṅ*), through abhihitatva, mark the relations between words. Now we consider the nominal suffixes. They fall under two categories.
   (a) vibhakti indicating a kāraka relation
       This marks a relation between a noun and a verb known as a kāraka relation. Sanskrit uses seven case suffixes to mark six kāraka

[7] tiṅkṛttaddhitasamāsaiḥ parisaṁkhyānam (ma. bhā. 2.3.1. vā.)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

relations viz. *kartā*, *karma*, *karaṇam*, *sampradānam*, *apādānam* and *adhikaraṇam*. The genitive suffix, in addition to marking a kāraka relation[8], is predominantly used to mark the noun-noun relation. There is no one-to-one mapping between the case suffixes and the kāraka relations, which makes it difficult to determine the relation on the basis of vibhakti alone.

(b) upapada vibhakti

In addition to the noun-noun relations expressed by the sixth case, there are certain words, most of them indeclinables called upapadas, which also mark a special kind of noun-noun relation. These indeclinables, mark a relation of a noun with an another noun, and in turn demand a special case suffix for the preceding noun. For example, the upapada 'saha' demands a third case suffix for the preceding noun as in

(10) *rāmeṇa saha sītā vanaṁ gacchati.*

3. Indeclinables (avyaya)

The indeclinables mark various kinds of relations such as negation, adverbial(manner adverbs only), co-ordination, etc. Sometimes they also provide information about interrogation, emphasis, etc. We distinguish the upapadas from the avyayas, mainly because, though most of the upapadas are also indeclinables, they demand a special case suffix on the preceding word, whereas it is not so with indeclinables.

For example, the relation of 'na' with 'gacchati' in the sentence

(11) *rāmaḥ gṛhaṁ na gacchati.*

is that of 'negation(niṣedha)'. Similarly, the relation of 'mandam' with 'calati' in the sentence

(12) *rāmaḥ mandaṁ calati.*

is that of 'adverbial(kriyāviśeṣaṇa)'. The relation of 'eva' with 'rāma' in the sentence

(13) *rāmaḥ eva tatra upaviṣṭati.*

is that of 'emphasis(avadhāraṇa)'.

4. Samānādhikaraṇa

Agreement in gender, number and case suffix marks *samānādhikaraṇa* (having the same locus), or the modifier-modified relation between two nouns as in

---

[8] kartṛkarmaṇoḥ kṛti (2.3.65)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

(14) *śvetaḥ aśvaḥ dhāvati.*
(15) *aśvaḥ śvetaḥ asti.*

In (14) as well as (15), the words *aśvaḥ* and *śvetaḥ* have the same gender, number and vibhakti indicating samānādhikaraṇa. However, there is a slight difference between the information being conveyed. In (15), the word *śvetaḥ* is a predicative adjective (vidheya viśeṣaṇa), while in (14) it is an attributive adjective.

2.3 Explicit Versus Implicit relations

Relations need not always be encoded directly through suffixes or morphemes. Sometimes the information is coded in the 'Language Convention'. The sūtra

samānakartṛkayoḥ pūrvakāle (3.4.21)

states that the suffix *ktvā* is used to denote the preceding of two actions that share the same kartā. Then the question is what relation does *ktvā* suffix mark? - the relation of kartṛtva or the relation of pūrvakālīnatva? or both?

Bhartṛhari in vākyapadīyam states (3.7.81-82),

pradhānetayoryatra dravyasya kriyāyoḥ pṛthak
śaktirguṇāśrayā tatra pradhānamanurudhyate 3.7.81

pradhānaviṣayā śaktiḥ pratyayenābhidhīyate
yadā guṇe tadā tadvad anuktāpi prakāśate. 3.7.82

i.e., in case X is an argument of both the main verb as well as the subordinate verb, it is the main verb which assigns the case and the relation of X to the sub-ordinate verb gets manifested even without any other marking.

From the sentences

(16) *rāmaḥ dugdhaṁ pītvā śālām gacchati.*
(17) *rāmeṇa dugdhaṁ pītvā śālā gamyate.*

it is clear that the vibhakti of *rāma* is governed by the main verb *gam*. And hence, the information that *rāma* is also the *kartā* of the verb *pā* is not expressed through any of the suffixes. The ktvā suffix expresses only the precedence relation (pūrvakālīnatva).

Similarly the sūtra

samānakartṛkeṣu (icchārtheṣu) tumun (3.3.158)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

states that in case of verbs expressing desire, the infinitive verb in the subordinate clause will have the same kartā as that of the verb it modifies. Here also the primary information available from the non-finite verbal suffix *tumun* is the relation of purpose.[9]

The sharing in case of *ktvā* and *tumun* suffixes is the result of the pre-conditions *samānakartṛkayoḥ* or *samānakartṛkeṣu* in 3.4.21 and 3.3.158 respectively which act as Language Conventions.

3 Factors useful for Śābdabodha

As mentioned above, the generation problem is a direct problem, and the analysis problem is a reverse problem, and is non-deterministic. This problem was well recognised by the mīmāṁsakas who proposed four conditions viz. ākāṅkṣā (expectancy), yogyatā (mutual compatibility), sannidhi (proximity) and tātparya (intention of the speaker) as necessary conditions for proper verbal cognition. With the help of examples, we explain below, how the first three factors play an important role in the rejection of non-solutions from among the several possibilities. We have not discussed the importance of the fourth factor, since the kind of analysis it involves is out of the scope of the present discussion.

3.1 *Ākāṅkṣā* (Expectancy)

In the sentence,
(18) *rāmaḥ vanaṁ gacchati.*

each of the 3 words in this sentence has multiple morphological analyses.
rāmaḥ = rāma {gender=m, case=1, number=sg},
= rā[10] {lakāra=laṭ, person=1, number=pl, voice=active, parasmaipadī}.

vanaṁ = vana {gender=n, case=1, number=sg},
= vana {gender=n ,case=2, number=sg}.

gacchati = gam {lakāra=laṭ, person=3, number=sg, voice=active, parasmaipadī},
= gacchat (gam śatṛ) {gender=m, case=1, number=sg},
= gacchat (gam śatṛ) {gender=n, case=1, number=sg}.

This may lead to the following two possible sentential analysis:
– *rāma* = kartā of the action indicated by *gam*,
vana = karma of the action indicated by *gam*.

[9] tumunṇvulau kriyāyām kriyārthyāyām (3.3.10)
[10] *rā* in the sense of *dāne* from the second (*adādi*) gaṇaḥ

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

– *vana* = karma of the action indicated by *rā*,
gacchati = simultaneity of the actions indicated by *rā* and *gam*,
vayam = kartā of the action indicated by the verb *rā* (not expressed explicitly, but through the verbal suffix).[11]

Of these two analysis, the second analysis can be ruled out on the basis of non-fulfilment of kartā and karma expectancics of the verb *gam*, and the sampradānam expectancy of the verb *rā*. The first analysis being complete in itself, it is preferred over the second one.

3.2 *Yogyatā* (Compatibility)

Consider the sentence,
(19) *śakaṭaṁ vanaṁ gacchati.*
The possible morphological analyses of each of the three words are given below.
śakaṭam = śakaṭa {gender=n, case=1, number=sg},
= śakaṭa {gender=n, case=2, number=sg}.

vanaṁ = vana {gender=n, case=1, number=sg},
= vana {gender=n, case=2, number=sg}.

gacchati = gam {person=3, lakāra=laṭ, number=eka, voice=active, parasmaipadī},
= gacchat (gam+śatṛ) {gender=m, case=1, number=sg},
= gacchat (gam+śatṛ) {gender=n, case=1, number=sg}.

Now, more than one word can't have the same kāraka role unless it is already expressed (abhihita). This leads to the following possible sentential analyses[12]:

– *śakaṭa* = kartā of the action indicated by *gam*,
vana = karma of the action indicated by *gam*.

– *vana* = kartā of the action indicated by *gam*,
śakaṭa = karma of action indicated by *gam*.

– *vana* = kartā of the action indicated by *gam*,
śakaṭa = modifier of *vana*.

– *vana* = karma of the action indicated by *gam*,
śakaṭa = modifier of *vana*.

[11] The sentence is interpreted as - (tasmin) gacchati (sati), vayaṁ vanaṁ rāmaḥ
As (he) goes, let us give the forest (to somebody).
[12] Assuming that the modifier is to the left, which need not be true in case of poetry.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

Out of these, the last two do not fulfill all the mandatory expectancies of a verb. Among the first two, the first one is preferable over the second one, since *śakaṭa* has an ability to move while *vana* can not move. Hence *śakaṭa* is preferable as a kartā of the verb *gam* than *vana*. Thus the yogyatā or the competency of the nouns to be eligible candidates for the kāraka roles plays an important role here. However, the context may overrule the condition of yogyatā. It is possible to have a reading where, all the residents of vana are going to see the new śakaṭa, and thus *vana* qualifies to be a kartā. The yogyatā and the context thus compete with each other and hence one needs discourse analysis to prune some of the possibilities.

3.3 *Sannidhi* (Proximity)

Consider,
(20) *rāmaḥ dugdham pītvā śālām gacchati.*
Here the possible analyses are:

- *rāma* = kartā of *gam*,
  dugdha = karma of *pā*,
  śālām = karma of *gam*,
  pā = preceding action with respect to *gam*.

- *rāma* – kartā of *gam*,
  dugdha = karma of *gam*,
  śālām = karma of *pā*,
  pā = preceding action with respect to *gam*.

A competent speaker rules out the second solution on account of non-compatibility of the arguments viz. *dugdha* and *śālā* do not have semantic competence to be the karma of *gam* and *pā* respectively.

The arguments in the correct solution are closer. We mark the words by their positions, and define the proximity measure of a relation as the distance between its two arguments, and the proximity measure of a solution as the sum of the proximity measures of the various relations in the parse. The proximity measure of the above two parses is

- *rāma* = kartā of *gam*
  (dist = position of *gam* - position of *rāma* = 5 -1 = 4)
  dugdha = karma of *pā* (dist = 3-2 = 1)
  śālām = karma of *gam* (dist = 5-4 = 1)
  pā = preceding action with respect to *gam* (dist = 5-3 = 2)
  Thus the total distance = 4 + 1 + 1 + 2 = 8

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

– *rāma* = kartā of *gam* (dist = 5-1 = 4)
dugdha = karma of *gam* (dist = 5-2 = 3)
śālām – karma of *pā* (dist = 4 3 = 1)
pā = preceding action with respect to *gam* (dist = 5-3 = 2)
Thus the total distance = 4 + 3 + 1 + 2 = 10

The one with greater proximity (or smaller distance) is preferred as a solution. Though Sanskrit is a free-word-order language, the following sentence with exchange of the karmas is not acceptable.

(21) **rāmaḥ śālām pıtva dugdhaṁ gacchatı.*

Equally unacceptable prose orders are

(22) **rāmaḥ pītvā śālām dugdhaṁ gacchati.*
(23) **rāmaḥ dugdham śālām pītvā gacchati.*

which involve crossing of links expressing the relations. A small pilot study of anvaya of Saṁkṣepa Rāmāyaṇa (Kutumbashastri, 2002) sentences show no evidence of crossing of links.

It is worth exploring the Calder mobile model suggested by Staal (1967) and further worked out by Gillon (1993) in the light of the mīmāṁsā principle of sannidhi. It may result in a better computational criterion for sannidhi.

4 Design Principles

The foregoing discussions lead to the following design principles for the constraint-based parser.

1. The relations will be marked as kāraka relations.
   [Using these kāraka relations and extra-linguistic knowledge, the semantic analysis may be carried out in the next level of processing.]
2. Only those relations that are marked directly by the morphemes will be extracted.
   [No relations that require some post-processing, or are based on secondary information will be extracted in the first step. The next level of processing will use this information to mark the unspecified or shared relations, if any.]
3. To prioritize the solutions, only the conditions of ākāṅkṣā and sannidhi will be used.
   [The condition of yogyatā will be used as and when the information is available in machine usable form, with the understanding that this knowledge may not be relied on completely.]
4. While dealing with prose, it will be assumed that there is no cross-linking of the relations between the words.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri

5 Mathematical Model

Let each word in a sentence be represented as a node in a graph, and the nodes be connected by the directed labelled edges. Then the problem of parsing a sentence may be modelled as

> Given a Graph G with n nodes, the task is to find a sub-graph T which is a directed Tree.[13]

Assuming that the words can be partitioned into two classes viz. the words which have an expectancy called demand words and the words which satisfy the demand called source words, Bharati et. al. reduced the parsing problem to matching a bipartite graph (Bharati,1995; 96). But in reality, the words can not be partitioned into two classes. We come across words which can be demand words in some context and source words in some other context, or in the same context a kṛdanta (primary derivative), e.g. can be both a demand word as well as a source word. Bharati et. al. (1995; 91) also needed the requirement of kārakas and their optionality for each verb. But then, a parser based on such information will fail to parse sentences with ellipsis, or the real corpus where we come across sentences with incomplete information. ,

With a robust parser, that produces at least partial solution in case of ellipsis, as an aim, we relax the above conditions. So we give away the constraint that a word can be exclusively either a demand or a source word. Further we treat all kārakas at the same level, irrespective of whether they are mandatory or optional, and assign penalty to lower the priority of those solutions which do not satisfy the mandatory expectancies.

We divide the problem into three parts:

1. For a given sentence, draw all possible labeled directed edges among the nodes.
2. Identify a sub-graph T of G such that T is a directed Tree which satisfies the given constriants.
3. Prioritize the solutions, in case there is more than one possible directed Tree.

In what follows we describe our model.

A matrix is a convenient way of representing the graphs for computing purpose. In our case, each word represents a node of a graph, and with each pair of nodes is associated zero or more labels, indicating the possible relations between these nodes. The strong constraint on these relations is that there can be at the most one label associated with a pair of nodes. This then naturally suggests a 3D matrix representation, whose elements are either 0 or 1, where the 3 dimensions represent two nodes and a relation label. Further, each word has one

[13] A tree is a graph in which any two vertices are connected by exactly one simple path.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

or more morphological analyses. Hence, corresponding to each node there exists a record with one or more cells, each cell representing one morphological analysis of the word. Let the j^{th} analysis of the i^{th} node be represented by $[i, j]$. Thus the address of a typical element of the 3D matrix is $([i, j], R, [l, m])$. The first pair of letters i and j correspond to the source word analysis, while the second pair of letters l and m represent the demand word analysis. R is the name of the relation of the l^{th} word to the i^{th} word. j indicates the morphological analysis of the i^{th} word responsible for this relation, and m indicates the morphological analysis of the l^{th} word that triggers this relation. In short the tuple $([i, j], R, [l, m])$ represents a relation R due to the m^{th} morphological analysis of the l^{th} word to i^{th} word due to its j^{th} morphological analysis. For ease of representation, we represent the tuple as (i, j, R, l, m). Thus, the initial graph with all possible relations between various nodes is represented as 5D matrix C such that
$C[i, j, R, l, m] = 1$, if such a relation exists,
$= 0$, otherwise.

Task 1: Based on the available information in a given sentence in the form of abhihitatva, vibhakti, sāmānādhikaraṇya, and the expectancies the matrix C is populated with 0s and 1s.

Here are sample rules (just enough to illustrate an example), expressed in English.

Rule 1:
If the sentence has
a noun(say 's') in prathamā vibhkati,
a verb(say 't') in kartari prayogaḥ, in 3rd person, and
's' and 't' agree in number,
then 's' is possibly a kartā of 't'.

Rule 2:
If the sentence has
a noun(say 's') in dvitīyā vibhkati,
a verb(say 't') in kartari prayogaḥ, and is sakarmaka (roughly transitive)
then 's' is possibly a karma of 't'.

Rule 3:
If the sentence has
a noun(say 's') in saptamī vibhkati, and
a verb(say 't'),
then 's' is possibly an adhikaraṇa of 't'.

Now consider the sentence
(24) *rāmaḥ vanaṁ gacchati.*

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri

The analyses of various words are numbered as follows:
[1, 1]: rāma {gender=m, case=1, number=sg},
[1, 2]: rā {gaṇaḥ=*adādi*, lakāra=laṭ, person=1, number=pl, prayogaḥ=kartari, parasmaipadī}.

[2, 1]: vana {gender=n, case=1, number=sg},
[2, 2]: vana {gender=n ,case=2, number=sg}.

[3, 1]: gam {lakāra=laṭ, person=3, number=sg, voice=active, parasmaipadī},
[3, 2]: gacchat (gam śatṛ) {gender=m, case=1, number=sg},
[3, 3]: gacchat (gam śatṛ) {gender=n, case=1, number=sg}.

The above 3 rules with this input then produce the following output showing all possible relations between various analses:
[2, 2] is a possible karma of [3, 2]
[2, 2] is a possible karma of [3, 3]
[2, 2] is a possible karma of [1, 2]
[2, 2] is a possible karma of [3, 1]
[2, 1] is a possible kartā of [3, 1]
[1, 1] is a possible kartā of [3, 1]

The resulting graph is shown in Figure 2.

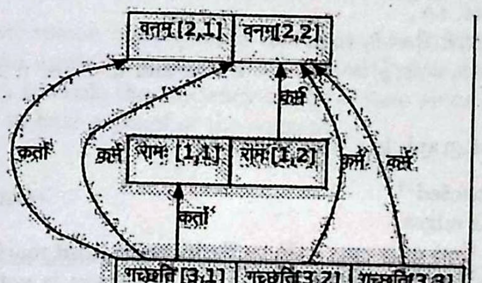


Fig. 2. Graph showing all possible relations

Task 2: In order to get a Tree from this graph, we impose the following constraints.

1. A morpheme(vibhakti) marks only one relation.
   I.e., a node can have one and only one incoming arrow.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

$\sum_{j,R,k,l} C[i,j,R,k,l] = 1, \forall i.$

2. Each kāraka relation is marked by a single morpheme.
   There can not be more than one outgoing arrow with the same label from the same cell, if the relation marks a kāraka relation,[14] i.e. there can not be two words satisfying the same kāraka role of the same verb.
   $\sum_{i,j} C[i,j,R,k,l] = 1$; for each tuple (R,k,l).

3. A morpheme does not mark a relation to itself.
   A word can't satisfy its own expectancy. i.e. a word can't be linked to itself[15]. Or there can not be self loops in a graph.
   $\sum_{j,R,k} C[i,j,R,i,k] = 0, \forall i.$

4. Only one valid analysis of every word per solution
   (a) If a word has both an incoming arrow as well as an outgoing arrow, they should be through the same cell.
   $\forall i \forall j \sum_{R,l,n} C[i,j,R,l,n] + \sum_{a,b,R,kl=j} C[a,b,R,i,k] \leq 1.$

   (b) If there is more than one outgoing arrow through a node, then it should be through the same cell.
   if, for some i,j,R,l,m C[i,j,R,l,m] = 1,
   then $\forall a \forall b \forall R \sum_{a,b,R,kl=j} C[a,b,R,l,k] = 0.$

5. All the words in a sentence should be connected.[16]

6. There are no crossing of links
   If all the nodes are plotted in a straight line, then they should not intersect each other. i.e.,
   if $C[i,j,R,k,l] = 1$, then
   $\forall v \forall y C[u,v,w,x,y] = 0$, if i < x < k and u < i or u > k.

The resultant graph is a Tree provided:

1. It is connected[17].
2. It has n-1 edges.
   The fact that only sup / tiṅ suffix in every word marks a relation with some other word in a sentence, and abhihita kāraka is not expressed by any sup suffix, it is guaranteed that there are exactly n-1 edges.

---

[14] adhikaraṇam is treated as an exception since one can have more than one adhikaraṇam as in
rāmaḥ adya pañca vādane gṛham agacchat.

[15] in case of some of the taddhita suffixes which are in svārtha, there will be self loops. But we do not consider the meaning of taddhita suffixes in the first step, and thus can avoid the self loops

[16] This condition is not yet implemented.

[17] Since, this condition is not yet implemented, the resulting graph need not be a Tree.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri

Task 3: The solutions are prioritized using the conditions specified below. For each of the solutions, the cost is calculated as
Cost = $\sum_{i,R,j} c_{iRj}$, where
i) $c_{iRj} = |j - i| * wt_R$, if $C[i, a, R, j, b] = 1$ for some a and b.
$= 0$ otherwise.
ii) $wt_R = rank(R)$, if R is a kāraka relation (appendix I shows the ranking)
$= 100$, otherwise.
This cost ensures the following:

1. ākāṅkṣā (kāraka relation) is preferred over other relations (rank[18] of the relations takes care of this.).
2. The ranking of the solutions on the basis of distance-based weights takes care of sannidhiḥ.

6 Implementation

The first task demands the inputs from grammar, whereas the second and the third tasks are purely mathematical ones, which can be handled by a constraint solver. The separation of tasks into three sub-tasks makes it not only modular, but also easy for a grammarian to test his/her rules independently. For the first task, an expert shell CLIPS is being used, whereas for the second task, a constraint solver MINION is being used. The system is available at

`http://sanskrit.uohyd.ernet.in/~anusaaraka/sanskrit/MT/test_skt.html`

There is no specific reason behind using these special software tools except the familiarity and the availability under the General Public License. No special efforts were put in towards the efficiency of the system since the main purpose of this exercise is to have a proof of the concept.

7 Performance

The current system allows only *padaccheda-sahita-eka-tiṅ-gadya-vākyam*. To measure the performance of this parser, we used hand tagged data. Around 110 sentences with single finite verb were selected from a school book (see appendix A for a sample). These sentences were tagged manually showing the relation of each word in the context. The sentences being simple, each sentence had a single possible parse in the context. There were 525 token words. The average length of the sentences was approximately 5, with a maximum length of 14 words. Morphological analyser is a pre-requisite for a parser. In order to avoid the cascading effect of errors due to non-availability of the morphological analysis, before running the parser, we ensured that the correct morphological analysis of all the words is being produced. Thus, given all possible correct analyses of the words, the task of the parser was to come up with a correct

[18] Better ranking scheme needs to be developed to take care of default word order.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

42

parse. Though the parser produces multiple parses, for the evaluation purpose, we chose only the first parse. Among the 113 sentences, 97 (86%) sentences had the first parse correct and 16 (14%) sentences had one relation wrong. Out of these 16, 10 relations had wrong label, 3 had wrong attachments and 3 went wrong in both the label as well as attachments.

The analysis of wrong results showed that most of the wrong relations were due to non-availability of appropriate knowledge to make the fine-grained distinction. For example, manually tagged corpus makes a distinction between kāla-adhikaraṇa and deśa-adhikaraṇa, gauṇa and mukhya karma in case of dvi-karmaka (di-transitive) verbs, hetu and karaṇam, to name a few. Another cause of ambiguity was the verbs in the curādi (10^{th}) gaṇa. For most of the verbs in this class, the causative and non-causative forms are the same. This then leads to a wrong parse, since we also allow elipsis. In case there are n (> 1) adjectives, there can be more than one possible way these adjectives can group with the following noun. But we produce a single parse where the adjectives are linked as a chain with the rightmost adjective qualifying the noun directly. This chain just indicates a chunk, and the internal grouping of these adjectives and also their relation with the head noun is left to the user for interpretation.

A sample output of a sentence

balyakale ramaḥ daśarathasya ajnayā vıśvāmitrasya yajñam rākṣasebhyaḥ rakṣitum vanam agacchat.

is produced in Figure 3.

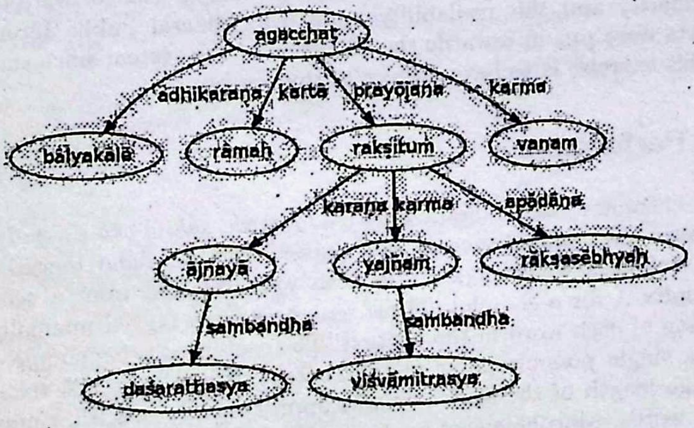


Fig. 3. Sample parse output

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

8 Challenges

The result with limited test cases is encouraging. The real corpus, even with small children's stories involves much more complex constructions, not necessarily confining to 'eka tiṅ vākyam'. The constructions involve co-ordination between two or more verbs, sentence connectives such as 'yadā-tadā, yathā-tathā, atha, tasmāt', etc. Thus, even at the level of simple texts, one can not do away with discourse analysis.

Another important problem that needs to be addressed is to handle a little more semantics than can be handled with syntactico-semantic relations. For example, it would be desirable to distinguish between hetu and karaṇa at least, though not between mukhya karma and gauṇa karma.

Third problem is regarding the upapadas. Upapada acts more like a function word (dyotaka) than a content word (vācaka). So in case of upapadas, it would be desirable to group the upapada together with the content word in the vibhakti it demands and then mark its relation with other content word. Thus e.g. in the sentence *rāmaḥ muninā saha vanaṁ agacchat*, it is desirable to parse it as in figure 4 than as in figure 5. This means a upapada should be treated as a function word, and as such should not be represented by a node.

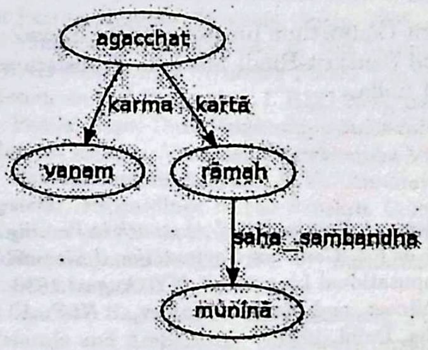


Fig. 4. saha-function

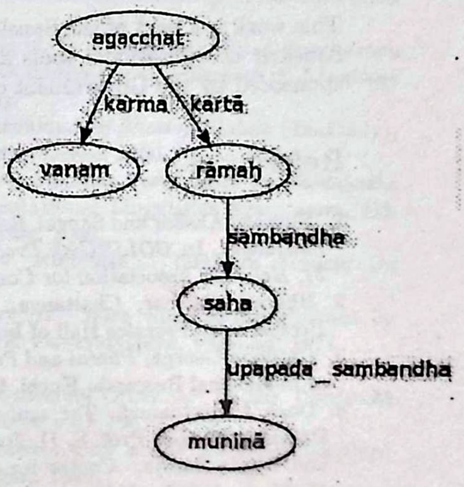


Fig. 5. saha-content

The vibhaktis, as we know, denote more than one meaning. For example, the second case suffix denotes the meaning of *kriyāviśeṣaṇa* (manner), *kāla* (time)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

or *adhvan* (path) in addition to the *karma*. To decide an appropriate role, now what one requires is the knowledge of yogyatā. In other words, our e-dictionaries should be rich with semantic properties of the words such as whether it denotes time, path or the manner, etc.

Since the parser does the analysis 'mechanically', it detects the problems of 'violation' of the rules more easily. We give just one example (more examples can be found in Gillon, 2002) from the anvaya of 'Saṁkṣepa Rāmāyaṇam'.

> guhena lakṣmaṇena sītayā ca sahitaḥ rāmaḥ vanena vanaṁ gatvā bahūdakāḥ nadīḥ tīrtvā bharadvājasya śāsanāt citrakūṭam anuprāpya vane ramyam āvasathaṁ kṛtvā devagandharvasaṅkāśāḥ te trayaḥ ramamāṇāḥ sukhaṁ nyavasan. (Śloka 30-32)

This sentence poses the following problems:
a) Whom does the phrase 'te trayaḥ' refer to?
b) *rāmaḥ* does not agree with the finite verb *nyavasan*. Is it not a violation of *samānakartṛkayoḥ pūrvakāle*?
c) Does *gatvā* precede *tīrtvā* or *nyavasan*?
d) In case of *vanena vanaṁ* what should be the meaning of the third case?

In spite of these problems, this parser can act as a tool to discover various kinds of semantic knowledge necessary to build a semantic parser.

9 Acknowledgement

This work is a part of the Sanskrit Consortium project entitled 'Development of Sanskrit computational tools and Sanskrit-Hindi Machine Translation system' sponsored by the Government of India.

References

1. Bharati, Akshar and Sangal, Rajeev, *A Karaka Based Approach to Parsing of Indian Languages*, In *COLING90: Proc. of Int. Conf. on Computational Linguistics (Vol. 3), Helsinki*, Association for Computational Linguistics, NY, August 1990.
2. Bharati, Akshar, Chaitanya, Vineet and Sangal, Rajeev, *NLP A Paninian Perspective*, Prentice Hall of India, Delhi, 1994.
3. Cardona George, *Pāṇini and Pāṇinīyas on śeṣa Relations*, Kunjunni Raja Academy of Indological Research, Kochi, 2007.
4. Dash Achyutanand, *The syntactic role of adhi in the Pāninian Kāraka system* in Pāṇinian Studies Prof. S. D. Joshi Felicitation volume, ed. Deshpande Madhav M, and Bhate Saroja, Center for South and Southeast Asian Studies, University of Michigan, U.S.A., 1991.
5. Gent, Ian P., Jefferson, Chris and Miguel, Ian. *MINION: A Fast, Scalable, Constraint Solver,* The European Conference on Artificial Intelligence 2006 (ECAI 06).

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

6. Gillon Brendan S. *Word Order in Classical Sanskrit* Indian Linguistics, v.57, n.1, pp. 1-35, 1996.
7. Gillon Brendan S. *Bhartṛhari's rule for unexpressed kārakas: The problem of control in Classical Sanskrit* Indian Linguistic Studies, Festschrift in Honor of George Cardona, Ed. Deshpande, Hook, Motilal Banarasidass, Delhi, 2002.
8. Hellwig, Oliver, *Extracting Dependency Trees from the Sanskrit Texts* Proceedings of the Sanskrit Computational Linguistics Symposium, Ed. Kulkarni and Huet, LNAI 5406, Springer Verlag, 2009.
9. Huet, Gérard, Formal Structure of Sanskrit Text: Requirements Analysis for a Mechanical Sanskrit Processor Proceedings of tho Sanskrit Computational Linguistics Symposium, Ed. Huet, Kulkarni and Sharf, LNAI 5402, Springer Verlag, 2009.
10. Huet, Gérard, Shallow syntax analysis in Sanskrit guided by semantic nets constraints Proceedings of International Workshop on Research Issues in Digital Libraries, Ed. Majumder, Mitra and Parui, ACM Digital Library, Dec 2006.
11. Jigyasu, Brahmadatt, 1979. *Ashtadhyayi (Bhashya) Prathamavrtti, three volumes,* Ramlal Kapoor Trust Bahalgadh, (Sonepat, Haryana, In dia) (In Hindi)
12. Joshi, S.D. (editor) 1968. *Patanjali's Vyakaraṇa Mahabhashya,* (several volumes), Univ. of Poona, Puno, India.
13. Joshi, S.D. and Roodebergen J.A.F., 1998. *The Aśṭādhyāyīof Pāṇini* (several volumes), Sahitya Akademi, Delhi, India.
14. Kiparsky, P. *On the Architecture of Panini's Grammar*, Proceedings of the Sanskrit Computational Linguistics Symposium, Ed. Huet, Kulkarni and Scharf, LNAI 5402, Springer Verlag, 2009.
15. Kutumbashastri, V. *Saṁkṣepa Ramāyaṇam*, Teach Yourself Samskrit series, ed., Rashtriya Sanskrit Samsthanam, New Delhi, 2002.
16. Lin D. *Dependency-based evaluation of MINIPAR.* In Workshop on the evaluation of Parsing Systems, Granada, Spain, 1998.
17. Marneffe M., MacCartney B. and Manning C. D. *Generating Typed Dependency Parses from Phrase Structure Parses* The fifth international conference on Language Resources and Evaluation, L REC 2006, Italy.
18. Pande, Gopal Dutt *Vaiyākaraṇa Siddhāntakaumudī of Bhattojidikshita* (Text only), Reprint Edition. Varanasi: Chowkhamba Vidyabhavan, 2000.
19. Ramakrishnamacharyulu, K.V. *Annotating Sanskrit Texts based on Śābdabodha systems,* Proceedings of the Sanskrit Computational Linguistics Symposium, Ed. Kulkarni and Huet, LNAI 5406, Springer Verlag, 2009.
20. Ramanujatatacharya, N.S. *Śābdabodha Mīmāṁsā* Institute Francis De Pondicherry, 2005.
21. Sharma, Raghunath *Vākyapadīyam, Part III* With commentary Prakāśa by Helaraja and Ambakartri Varanaseya Sanskrit Visvavidyalaya, Varanasi, 1974.
22. SK: Siddhāntakaumudī See Pande
23. Staal, J.F. *Word Order in Sanskrit and Universal Grammar* Reidal, Dordercht (Foundations of Language, Supplementary series: v.5), 1967.
24. Sleator D. D., Temperley D. *Parsing English with a link grammar* In third international Workshop on Parsing Technologies, 1993.

A Sample story

nadyāḥ taṭe ekaḥ vṛkṣaḥ asti| vṛkṣasya samīpam ekā śilā asti| vṛkṣasya śākāsu nīḍāḥ santi| nīḍeṣu vihagāḥ vasanti| nīḍāḥ vihagān rakṣanti| vṛkṣasya

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

adhaḥ vānarāḥ santi| kapayaḥ gṛham na racayanti| te sarvadā itastataḥ bhramanti| ekasmin divase śītam tān pīḍayati| te śītāt trāṇāya agnim icchanti| kutrāpi te agnim na vindanti| ekaḥ guñjāyāḥ phalāni paśyati| guñjāyāḥ phalāni raktāni santi| saḥ agneḥ sadṛśāni guñjāyāḥ phalāni ānayati| tāni guñjāyāḥ phalāni śilāyām saṁharati| te sarve guñjā-phalam paritaḥ upaviśanti| agneḥ icchayā te mukhaiḥ tāni dhamanti| te agnim na vindanti| te vānarāḥ analāya vṛthā āyāsam kurvanti| teṣām śītam na naśyati| kapayaḥ mūrkāḥ santi|

B Relations

The relations used, along with their ranks are given in Table 1.

(0)	upapada vibhakti	(12)	kāla-adhikaraṇaṁ
(1)	kartā	(13)	viṣaya-adhikaraṇaṁ
(2)	prayojaka kartā	(14)	kartā-samānādhikaraṇaṁ
(3)	prayojya kartā	(15)	viśeṣaṇaṁ
(4)	karma	(16)	kriyā-viśeṣaṇaṁ
(5)	reserverd for gauṇakarma	(17)	tādarthya
(6)	reserverd for mukhyakarma	(18)	pūrvakālīna
(7)	karaṇaṁ	(19)	sambandha
(8)	sampradānaṁ	(20)	kāraka ṣaṣṭhī
(9)	apādānaṁ	(21)	niṣedha
(10)	adhikaraṇaṁ	(22)	sambodhana
(11)	deśa-adhikaraṇaṁ		

Table 1. Relations

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation, Delhi and eGangotri.

संस्कृतार्थं प्रतिबन्धबहुलाश्रितवाक्यविश्लेषकम्

(Designing a Constraint Based Parser For Sanskrit)

अम्बा कुलकर्णी, शीतल पोकार, देवानन्द शुक्ल
संस्कृताध्ययनविभागः, हैदराबाद-विश्वविद्यालयः
भाग्यनगरम् (हैदराबाद) भारतम्
apksh@uohyd.ernet.in, sjpokar@gmail.com, dev.shukl@gmail.com

सारांशः - शाब्दबोधप्रक्रियायां प्रत्येक-शब्दस्य अन्येन शब्देन सह अन्वयः आवश्यकः। सामान्यतः एतादृशं ज्ञानं व्याकरणेन संभवति। क्रियात्मकव्याकरणं विश्लेषयति यत् भाषायां पदार्थानां कथं सम्बन्धाः उपस्थापिताः भवन्ति। पाणिनीयव्याकरणं वक्तृपक्षतः साधुवाक्यस्य निर्माणप्रक्रियां सूचयति। श्रोतुः तस्य अवबोधनं कथं भवति इति साक्षात् तत्र चर्चा न कृता। यस्य कस्यचित् व्याकरणस्य प्रतिबन्धबहुलाश्रितकारकविश्लेषकस्य निर्माणे आर्थिकदृष्ट्या भाषायामुपस्थितानां शब्दसम्बन्धानां शुद्धरूपेण ज्ञानमपेक्षितम्। तत्र मुख्याः त्रयः - कः व्यक्तसम्बन्धः, कः प्रकटयति व्यक्तसम्बन्धम्, केन प्रकारेण व्यक्तः स सम्बन्धः इति। एतेन ज्ञानेन निर्माणसिद्धान्तस्योपस्थितिर्भवति। येन सैद्धान्तिकरूपेण कारकविश्लेषकस्य (Parser) निर्माणं भवति। आकाङ्क्षामाधारीकृत्य कारकविश्लेषकं वाक्यस्य सर्वविधविश्लेषणमुपस्थाप्य तेषु सान्दर्भिकशुद्धसम्बन्धानां चयनं कृत्वा अन्यानशुद्धसम्बन्धान् त्यजति। पुनश्च तेषां श्रेणीबन्धनम् सन्निध्याधारेण करोति। प्रायोगिकदृष्ट्या एतस्य परीक्षणं कृतमस्माभिः परिणामोऽपि समीचीनः प्रतिभाति।

परिचयः

शब्देभ्यः जायमानः यः बोधः सः शाब्दबोधः। शाब्दबोधे 'कः मुख्यविशेष्यः?' विषयेऽस्मिन् न्याय-व्याकरण-मीमांसाशास्त्रेषु वैमत्यं दरीदृश्यते। मुख्यविशेष्यविषये वैमत्ये सत्यपि प्राथमिकरूपेण वाक्ये स्थितानां पदानां परस्परसम्बन्धानां विषये सर्वेषाम् ऐकमत्यमेव। तद् वाक्यं एकं पदमपि भवितुमर्हति उत सम्पूर्णग्रन्थोऽपि। अत्र 'एकतिङ् वाक्यम्' इत्येव सर्वत्र वाक्यत्वेन स्वीकुर्मः।

भाषानियमानाधारीकृत्य यत् संश्लेषणात्मकं व्याकरणं वर्तते तद् Generative Grammar इति नाम्नाभिधीयते। विश्लेषणार्थमिदं व्याकरणं वैपरीत्येन (वाक्यात् व्याकरणविश्लेषणपर्यन्तम्) द्रष्टव्यम्। विपरीतक्रमः कुत्रचित् सरलः एवञ्च निश्चयात्मकश्च भवति। 'व्यवकलनम्' (subtraction) योगार्थम् एका निश्चयात्मकविपरीतप्रक्रिया वर्तते। विपरीतप्रक्रिया सर्वदा निश्चायत्मिका भवितुं नार्हति। परिचयार्थं अनिश्चयात्मकविपरीतक्रमस्योदाहरणं पश्यामः।

पाणिनेः सूत्रद्वयमिदानीं प्रस्तूयते :

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

अनभिहिते (पा० 2..3. 1)

कर्तृकरणयोस्तृतीया (पा० 2. 3. 18)

उपर्युक्तसूत्रद्वयाधारेण अर्थः निष्पादितः अनभिहिते कर्तरि करणे च तृतीयाविभक्तिर्भवति।

(1) रामेण बाणेन बाली हन्यते।

इदानीम्, यदा श्रोता (यः व्याकरणं जानाति) एतद्वाक्यं श्रुत्वा चिन्तयति यदत्र कर्मणिप्रयोगः वर्तते। द्वौ शब्दौ तृतीयान्तौ स्तः। यदि सः न जानाति यत् 'राम' इति काचन व्यक्तिः एवं 'बाण' इति करणम् तर्हि समीचीनार्थज्ञाने सोऽसफलः भवति। भाषेतरव्यवहारज्ञानाभावात् तत्रार्थे सम्भावने द्वे स्तः। एका तु 'राम' इति कर्ता एवं 'बाण' इति करणम्। द्वितीया 'राम' इति करणम् एवं 'बाण' इति कर्ता। इत्येतदेव अनिश्चयात्मकविपरीतक्रमस्योदाहरणं सिद्ध्यति।

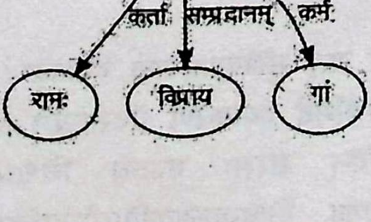
वाक्यस्थपदानां परस्परं सम्बन्धं सूचयति वाक्यविश्लेषकम्। भाषार्थं द्विधा वाक्यविश्लेषकस्य निर्माणं भवति। प्रथमं तावत् 'साङ्ख्यिकीयम्' एतत् यान्त्रिकशैल्या मानवकृतानुबन्धसङ्ग्रहेण यन्त्रशिक्षणप्रविधिद्वारा शिक्षितम् भवति। समाजे एतस्य प्रभुत्वं वर्तते। एतदर्थं मानवकृतानुबन्धितबृहत्सङ्ग्रहस्यावश्यकता भवति। द्वितीयन्तु 'व्याकरणाधारितम्' एतेन व्याकरणदृष्ट्या नियमैः असम्बद्धं निष्कास्य सम्भावितसमाधानमनुमीयते। विभिन्नभाषासु द्वाभ्यामाधाराभ्यां वाक्यविश्लेषकं निर्मितम्[1]। पाणिनीयव्याकरणमनुसृत्य आधुनिकभारतीयभाषार्थं भारतीत्यादिभिः[2] व्याख्या कृता। एतत् वाक्यविश्लेषकं द्विभागीयं रेखाङ्कनतोलनसमस्याम् (Bipartite graph matching problem) आधारीकृत्य निर्मितम्। हेल्विग (Hellwig - 2008) इति विदुषा 'साङ्ख्यिकीयम्' इत्याधारितं वाक्यविश्लेषकविषये चर्चा कृता। वेत् (Huet 2006, 2009) महोदयेन कर्मव्यवस्थामाधारीकृत्य रेखाङ्कनतोलनप्रक्रियाविधिना (Graph-matching algorithm) एकम् उत्तानवाक्यविश्लेषकम् (Shallow Parser) निर्मितम्। तस्य मुख्यं लक्ष्यम् वाक्ये स्थितानामसाधुसम्बन्धानां तिरस्कारः आसीत्।

संस्कृते व्याकरण-सम्बन्ध-संकेतनम् -

'वाक्यविश्लेषणम्' नाम वाक्ये विद्यमानशब्दान् वर्गीकृत्य तन्मध्ये विद्यमानसम्बन्धदर्शनम्। यथा -

रामः विप्राय गां ददाति।

उपर्युक्तोदाहरणं दृष्ट्वा अग्रिमचित्रवद् ज्ञानम् उपपद्यते।



वाक्यविश्लेषकं सम्बन्धानां

1 Lin, 1998; Marneffe, 2006; Sleator, 1993.

2 Bharati, et. Al (1995; 85-100)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

केन प्रकारेण विश्लेषणं कुर्यात्। वाक्यविश्लेषकनिर्माणार्थं त्रिधा विवेचनं आवश्यकम्- कः व्यक्तसम्बन्धः, कः प्रकटयति व्यक्तसम्बन्धम्, केन प्रकारेण व्यक्तः स सम्बन्धः इति। अत्र सम्बन्धः कीदृशः भवेत्? वस्तुतः शाब्दबोधे पदार्थं स्वीकृत्य अनन्तरं तस्य पदेन सह सम्बन्धं, तदनन्तरं पदस्य अन्येन पदेन सह सम्बन्धं स्थापयति। यथा उपर्युक्तोदाहरणे 'राम' इत्यस्य पदार्थः रामत्वधर्मविशिष्टः, एतेन बोध्यं 'राम' पदम्, तत्पदं क्रियया सह कर्तृत्वेनान्वेति। इदानीं 'राम' इति शब्दे रामत्वमपि वर्तते कर्तृत्वमपि वर्तते। 'रामत्वम्' इति पदार्थसम्बन्धः, कर्तृत्वम् इति कारकसम्बन्धः। व्यवहारज्ञानं यन्त्रे नास्ति अतः यन्त्रं सामान्यरूपेण कारकसम्बन्धं स्वीकृत्य कार्यं विदधाति।

संस्कृते व्याकरणसम्बन्धसङ्केतनम्

व्यवहारजगत् भिन्नं शब्दजगत् च भिन्नं भवति। व्यवहारे यद् अधिकरणं तस्य शब्दजगति कर्तृत्वेन विवक्षितत्वात् तत्र कर्तृत्वं भवति। यथा - स्थाली पचति। व्यवहारे स्थाली अस्ति अधिकरणं परन्तु वक्तुरिच्छया वाक्ये तस्य कर्तृत्वं वर्तते। अतः शब्दजगति विभक्त्या यः सम्बन्धः द्योत्यते तम् एव सम्बन्धं 'वाक्यविश्लेषकम्' स्वीकरोति। इदमेयोदाहरणान्तरैः विशदरूपेणत्र प्रतिपादयामः।

कः सम्बन्धः

'स्वतन्त्रः कर्ता' इति कर्तृसंज्ञाविधायकं पाणिनीयं सूत्रम्। सूत्रस्यास्य एतानि त्रीणि उदाहरणानि द्रष्टव्यानि -

देवदत्तः पचति।
स्थाली पचति।
काष्ठानि पचन्ति।

यद्यपि देवदत्तस्थालीकाष्ठानां पाकक्रियायां यथासंख्यं कर्तृत्वाधिकरणत्वकरणत्वानि प्रसिद्धानि लोके। परन्तु व्याकरणे त्वेतेषां सर्वेषां पाकक्रियायां कर्तृत्वं साध्यते। कथं तद् ? प्रतिकारकं क्रियाभेदात्। उक्तञ्च महाभाष्ये -

"कोऽसौ प्रतिकारकं क्रियाभेदः पचादीनाम् ?

अधिश्रयणोदकासेचनतण्डुलावपनैधोपकर्षणक्रियाः प्रधानस्य कर्तुः पाकः।[3]

अधिश्रयणोदकासेचनतण्डुलावपनैधोपकर्षणादिक्रियाः कुर्वन्नेव 'देवदत्तः पचति' इत्युच्यते। तत्र तदा पचिर्वर्तते। एषः प्रधानकर्तुः पाकः। एतत्प्रधानकर्तुः कर्तृत्वम्।

द्रोणं पचत्याढकं पचतीति सम्भवनक्रिया धारणक्रिया चाधिकरणस्य पाकः।

द्रोणं पचत्याढकं पचतीति सम्भवनक्रियां धारणक्रियां च कुर्वती स्थाली "पचति" इत्युच्यते। तत्र तदा पचिर्वर्तते। एषोऽधिकरणस्य पाकः। एतदधिकरणस्य कर्तृत्वम्।

एधाः पक्ष्यन्त्याविक्लित्तेर्ज्वलिष्यन्तीति ज्वलनक्रियाकरणस्य पाकः[4]।

एधाः पक्ष्यन्ति, आ विक्लित्तेर्ज्वलिष्यन्तीति ज्वलनक्रियां कुर्वन्ति काष्ठानि "पचन्ति"

---

3 वार्तिकम् - 8, कारके (1.4.23)

4 वार्तिकम् - 10, कारके (1.4.23)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

इत्युच्यन्ते। तत्र तदा पचिर्वर्तते। एष करणस्य पाकः। एतत्करणस्य कर्तृत्वम्।"

अत एव विभक्त्या केवलं कर्तृत्वस्याभिज्ञानाद् आर्थिसंबन्धमवगणय्य व्याकरणमनुसृत्य वाक्यविश्लेषकेनात्र देवदत्त-स्थाली-काष्ठानां केवलं कर्तृत्वं प्रदर्श्यते न तु अधिकरणत्वं करणत्वं वा।

शेषे (पा० 2.3.50) अनेन सूत्रेण स्वस्वामिभावादिसम्बन्धबोधनार्थं षष्ठीविभक्तिः।

उक्तादन्यः (कारकप्रातिपदिकाद्भिन्नः) शेषः भवति। तस्य सम्बन्धविवक्षायां षष्ठीविभक्तिर्विधीयते [5]। अस्य उदाहरणानि प्रदत्तानि -

वृक्षस्य शाखा।
पितुः पुत्रः।
राज्ञः पुरुषः।

अत्रापि यथासंख्यं अवयव-अवयवि-जन्य-जनक-स्व-स्वामि-संबन्धाः प्रसिद्धा लोके। किन्तु व्याकरणे सर्वे "षष्ठी शेषे" (पा०2.3.50) सूत्रान्तर्गतपरिगणिताः षष्ठ्या विभक्त्या सूच्यन्ते। विभक्तिं दृष्ट्वा सम्बन्धषष्ठ्यैवाभिज्ञानाद् सम्बन्धषष्ठ्येव विश्लेषकेन स्वीक्रियते। सम्बन्धविशेषस्याङ्कनं तु न करोति।

अधिशीङ्स्थासां कर्म (1.4.46) अधिपूर्वाणां शीङ् स्था आस् इत्येतेषाम् आधारः यः तत् कारकं कर्मसंज्ञकं भवति।

अत्र व्यवहारे वस्तुतः भिन्नकारकम् वर्तते परन्तु शब्दजगति भिन्नं कारकम्। स्पष्टीकरणार्थं सूत्रस्यास्योदाहरणानि प्रदत्तानि -

हरिः वैकुण्ठम् अधिशेते।
मुनिः शिलापट्टम् अधितिष्ठति।
साधुः पर्वतम् अध्यास्ते।

वैकुण्ठशिलापट्टपर्वताः यथासंख्यं हरिमुनिसाधूनामधिकरणानि लोके। किन्तु शब्दजगति "अधिशीङ्स्थासां कर्म" (पा० 1.4.46) सूत्रानुसारं तेषां कर्मत्वमङ्गीक्रियते। विभक्त्याऽत्र कर्मत्वमेव द्योत्यतेऽतः कर्म-संबन्ध एव प्रदर्श्यते विश्लेषकेन।

सहयुक्तेऽप्रधाने (2.3.19) इत्येते सहार्थेन युक्तेऽप्रधाने तृतीयाविभक्तिर्भवति। स्पष्टीकरणार्थं उदाहरणं पश्यामः-

मात्रा सह शिशुः अङ्के उपविश्य गृहम् आगच्छति।

वाक्येऽस्मिन् व्यवहारे माता एव आगमन क्रियां करोति, शिशुः तु अङ्के एव उपविष्टः। परन्तु शब्दजगति 'शिशु' मुख्यक्रियया (आगच्छति) सह अन्वेति अतः तस्य प्रधानत्वेन कारकत्वम्, अप्रधानत्वेन मातुः कारकत्वं नास्ति। अतः अप्रधानत्वात् सहयुक्ते तत्र तृतीयाविभक्तिः आगता।

---

5 शेषे षष्ठी (2.3.50)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

इत्थं व्यवहारे शब्दजगति यः सम्बन्धः, तयोः भेदे सति शब्दजगति यः सम्बन्धः स एव स्वीक्रियते। शाब्दबोधप्रक्रियायां वैयाकरणानां मतमेव वाक्यविश्लेषकनिर्माणे प्रमाणत्वेनाङ्गीकृतम्।

केन व्यक्तः सम्बन्धः

1) अभिहितत्वम् – उक्तं पाणिनिनाऽनभिहिते अनुक्ते, अनिर्दिष्टे कारके विभक्तिर्भवति। केनानभिहिते ? तिङ्कृत्तद्धितसमासैः अनभिहिते। अभिहिते तु प्रातिप्रदिकार्थमात्रे प्रथमा भवति। यथा –

➢ रामः वनं गच्छति।

अत्र "गच्छति" पदेन अभिहितो "रामः"। कथं ज्ञायते ? तिपा सामानाधिकरण्यात् रामस्य। तिप् इत्यनेन यत्सामानाधिकरण्यं रामस्य तत् शप् इत्यनेन प्रत्ययेन द्योत्यते। रामाभिन्नकर्तृको गमनानुकूलो व्यापारोऽत्र। तिपा अभिहितत्वात् राम-शब्दात् प्रथमा, यया रामस्य कर्तृत्वमभिज्ञायते।

तथैव –

➢ रामेण वनं गम्यते।

अत्र 'गम्यते' पदेन अभिहितं वनम्। कथं ज्ञातम् ? त-प्रत्ययेन वनस्य सामानाधिकरण्याद् ज्ञायते। तत् सामानाधिकरण्यं यक्प्रत्ययेन द्योत्यते। अभिहितत्वात् वन-शब्दात् प्रथमा, यया वनस्य कर्मत्वमभिज्ञायते।

➢ रामः ग्रामं गतः।

उक्तोदाहरणेऽत्र 'गतः' इति पदे वर्तमानः क्त-प्रत्ययः कर्तरि सद्भावात्, सामानाधिकरण्यात् रामं कर्त्तारमभिदधाति। क्तेन कर्त्रभिधानात् राम-शब्दात् प्रथमा, यया रामस्य कर्तृत्वमभिगम्यते।

➢ रामेण ग्रामः गतः।

'गतः' इत्यस्मिन् पदे क्त-प्रत्ययः कर्मणि वर्तते। अतः सामानाधिकरण्यात् क्त-प्रत्ययेनाऽत्र ग्रामरूपं कर्माभिधीयते। क्त-प्रत्ययान्तेनाभिहितत्वाद् ग्राम-शब्दादत्र प्रथमा, यया ग्रामस्य कर्मत्वमभिगम्यते।

2) विभक्तिः –

1. कारकविभक्तिः कारकसम्बन्धं दर्शयति। यथा – नृपः ब्राह्मणाय गां ददाति। अत्र गोब्राह्मणाभ्यां ये द्वितीयाचतुर्थ्यौ ते गोब्राह्मणयोः यथाक्रमं कर्मत्वं सम्प्रदानत्वञ्च दर्शयतः।

2. उपपदनिमित्तिका विभक्तिरुपपदविभक्तिरुच्यते। यथा – रामेण सह सीता वनं गच्छति। अत्र रामशब्दाद् या सहपदनिमित्तिका तृतीया सा उपपदविभक्तिरुच्यते। यया उपपदसम्बन्धः स्पष्टीक्रियते।

2. प्रतिपदोक्ता विभक्तिः -पाणिनिना उपपदविभक्तिभिन्ना प्रतिपदोक्तविहिता प्रतिपदोक्तविभक्तिरत्र। यथा – पादेन खञ्जः इत्यङ्गविकारे तृतीया, शिखया

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

परिव्राजक इति इत्थंभूतलक्षणातृतीया, धनेन कुलमिति हेतौ तृतीया, गवां गोषु वा कृष्णा सम्पन्नक्षीरतमेति निर्धारणे षष्ठी सप्तमी वा इत्यादयः प्रतिपदोक्तविभक्तय उच्यन्ते।

3. षष्ठीविभक्तिः -

संबन्ध-षष्ठी = शेषे षष्ठी विहिता "षष्ठी शेषे(पा॰ 2.3.50)" इति। कर्मादिभ्योन्यः प्रातिपदिकार्थव्यतिरिक्तः स्वस्वामिसंबन्धादिः शेषः, तत्र षष्ठी विभक्तिर्भवति, सा संबन्ध-षष्ठी अत्र। एकशतं षष्ठ्यर्थाः इति उक्तं महाभाष्ये। परन्तुः कः षष्ठ्यर्थः इति ज्ञानं व्यवहारेणैव भवति। व्यवहारज्ञानाभावात् षष्ठ्यर्थस्य निर्णयः असम्भवः। अतः यथा पाणिनिना सम्बन्धे षष्ठी विधास्यते तथैवात्रापि सम्बन्धमात्रं द्योत्यते।

कारक-षष्ठी = 'कर्तृकर्मणोः कृति(पा॰ 2.3.65)' इति कृद्योगे विहिता षष्ठी कारकषष्ठीति उच्यते। 'शब्दानाम् अनुशासनम्' , 'देवदत्तस्य पाकः' इति।

3) उपपदभिन्नाव्ययानि -

अव्ययान्यपि संबन्धदर्शने सहायकानि। यथा -

1) देवदत्तः ओदनं न भुङ्क्ते।
'न' अत्र निषेधदर्शकः।

2) रामः एव तत्र उपविशेत्।
बलाधानद्योतकोऽत्र 'एव'।

3) इच्छामि ज्ञातुं नरमेवं विधं हि मे परं कौतूहलम्।
इच्छतेः कारणस्य द्योतको 'हि' अत्र।

4) क्रियाविशेषण-पदानि -

धात्वर्थव्यापारजन्यफले अभेदान्वयात् क्रियया सामानाधिकरण्यं तस्माद्विशेषणत्वञ्च क्रियाविशेषणानाम्। तान्यपि द्विधानि अव्ययसुबन्तस्वरूपभेदत्वात्।

अव्ययस्वरूपानि -

➢ देवदत्तः सुष्ठु पचति।

देवदत्तकर्तृकपचौ सुष्ठुपदार्थस्याभेदान्वयात् क्रियया सामानाधिकरण्यम् विशेषणत्वञ्च।

सुबन्तस्वरूपाणि -

➢ रामः मन्दं चलति।
➢ अश्वः वेगेन धावति।
➢ रमा गीतं मधुरं गायति।
➢ स्तोकं पचति।

रामादिकर्तृको मन्दत्वाद्यभिन्नश्चलनाद्यनुकूलो व्यापारः सर्वत्र। अतः धात्वर्थव्यापारजन्यफले विभिन्नविभक्तिकमन्दादिपदार्थस्याभेदान्वयात् क्रियया सामानाधिकरण्यम् विशेषणत्वञ्च।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

5) समानाधिकरणत्वम् –

समानलिङ्गवचनविभक्तिकपदयोः पदानाञ्च समानाधिकरणत्वमर्थाद् तानि पदानि विशेष्यविशेषणभावं द्योतयन्ति | यथा –

➢ श्वेतः अश्वः अस्ति।

अत्र अश्वत्वश्वेतत्वयोः समानाधिकरणत्वाद् विशेष्यविशेषणभावसंबन्धः समानलिङ्वचनविभक्तित्वाद् ज्ञायते।

➢ अश्वः श्वेतः अस्ति।

अत्रापि अश्वत्वश्वेतत्वयोः समानाधिकरणत्वं समानलिङ्वचनविभक्तित्वञ्च। परमुद्देशविधेयभावमाश्रित्य श्वेतत्वमत्र विधेयविशेषणमिति।

केन प्रकारेण व्यक्तः सम्बन्धः –

न सर्वे संबन्धाः विभक्त्या द्योत्यन्ते किन्तु भाषाप्रवाहेणावगम्यन्ते। यथा –

➢ रामः दुग्धं पीत्वा शालां गच्छति।

अत्र गमनक्रियायां रामस्य कर्त्तृत्वं तिङा ज्ञायते। परं पिबतावपि रामस्य कर्त्तृत्वं विद्यते। तत् कथं ज्ञायते ? क्त्वा-प्रत्ययेन। लोके 'समानकर्तृकयोः पूर्वकाले(पा॰ 3.4.21)' समाने कर्तरि पूर्वकाले धात्वर्थे वर्तमानाद् धातोः क्त्वा विधीयते। अतोऽत्रापि पूर्वकालिकत्वात् पिबतेः क्त्वा-प्रत्ययः, तेन क्त्वाप्रत्ययेन ज्ञाप्यते यत् पिबतेर्गच्छतेश्च कर्त्ता राम एव इति। अर्थात् राम एव पूर्वं दुग्धं पिबति। ततश्च राम एव शालां गच्छति। इत्थमत्र पिबतौ रामस्य कर्त्तृत्वं विभक्त्या न द्योत्यते किन्तु भाषाप्रवाहेणावगम्यते अर्थात् एतादृक्षु स्थलेषु भाषायां क्त्वा प्रयुज्यते येन कर्त्तृत्वं ज्ञायते। उक्तं हरिणापि –

प्रधानेतरयोर्यत्र द्रव्यस्य क्रिययोः पृथक्।
शक्तिर्गुणाश्रया तत्र प्रधानमनुरुध्यते।। 3.7.81
प्रधानविषया शक्तिः प्रत्ययेनाभिधीयते।
यदा गुणे तदा तद्वदनुक्तापि प्रकाश्यते।।3.7.82

अर्थात् क्रियाद्वयस्य यदा एक एव कर्त्ता तदा विभक्त्या प्रधानक्रियया कर्ता अभिधीयते। यद्यपि गौणक्रियायामपि तस्य कर्त्तृत्वं भवति। तच्च गौणकर्तृत्वं विभक्त्या न बोध्यते, किन्तु प्रतीयते।

➢ रामः रोटिकां पक्त्वा ओदनं भुङ्क्ते।

➢ रामः पक्त्वा ओदनं भुङ्क्ते।

प्रथमवाक्ये कर्मद्वयस्य विद्यमानत्वात् स्पष्टं यत् रोटिकाकर्मकः पचिरोदनकर्मको भुजिरत्रेति। किन्तु, द्वितीये न तथा। तत्र त्वेक एव पदार्थ 'ओदनः' इति कर्मणि दृश्यते। क्त्वा-प्रत्ययस्य समानकर्त्तृद्योतकत्वं, न तु समानकर्मद्योतकत्वम्। अतः कर्मसाकाङ्क्षस्य पचेः किं कर्म इति पुनः स्मर्यते हरिरत्र "प्रधानविषया शक्तिः प्रत्ययेनाभिधीयते। यदा गुणे तदा तद्वदनुक्तापि प्रकाश्यते।। " इति पचावपि गौणक्रियायाम् ओदनस्य कर्मत्वं प्रधानक्रियाभुजिवत्।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

➢ धावन्तम् अश्वं पश्यति।

यद्यपि समानाधिकरणं धावनस्याश्वत्वस्य समानलिङ्वचनविभक्तित्वञ्च। तस्माद् विशेष्यविशेषणभावसंबन्धे सत्यपि धावने अश्वस्य कर्तृत्वम्। कथमभिज्ञायते ? 'कर्तरि कृत् (पा० 3. 4. 67)' इति शतुः कर्त्तरि सद्भावात् कर्त्तारमभिदधाति। कृदन्तत्वात् स्वयं विशेषणीभूतो 'लटः शतृशानचावप्रथमासमानाधिकरणे (पा० 3. 2. 124)' इति शतृप्रत्ययः कर्त्रा सामानाधिकरण्यात् विशेष्य-कर्तुः लिङ्वचनविभक्तिं गृह्णाति।

शाब्दबोधे सहकारिकारणानि

उपर्युक्तविश्लेषणेन ज्ञायते यत् शाब्दबोधे व्युत्पादनसमस्या एका साक्षात् समस्या वर्तते एवं विश्लेषणसमस्या विपरीतसमस्यापरा अनिश्चयात्मकसमस्या। मीमांसकाः एतासां समस्यानामुपरि ध्यानमददुः। निराकरणरूपेण आकाङ्क्षा योग्यता सन्निधिः तात्पर्यमित्येते सिद्धान्ताः उपस्थापिताः। 'आकाङ्क्षा योग्यता सन्निधिः' एते त्रयः सिद्धान्ताः असम्भावितसमाधानां निराकरणं कुर्वन्ति।[6] तात्पर्यविषयेऽस्माभिः वर्तमानविषयस्यासम्बद्धत्वादत्र चर्चा न कृता। एतेषां सिद्धान्तानां यन्त्रे उपयोगः कथम् एतत्सर्वं विशदरूपेणाग्रे निरूपितम् -

आकाङ्क्षा -

'रामः', 'वनं', 'गच्छति' - इति शब्दत्रये प्रतिशब्दम् एकाधिकं विश्लेषणं प्राप्नोति। यथा -

रामः - पुल्लिङ्गम्, प्रथमा, एकवचनम्
रा-धातुः, लट्, कर्तरि, उ.पु., बहुवचनम्
वनम् - नपुंसकम्, प्रथमा, एकवचनम्
नपुंसकम्, द्वितीया, एकवचनम्
गच्छति - गम्-धातुः, लट्, कर्तरि, प्र.पु., एकवचनम्
गम्-धातुः, शतृ, पु., सप्तमी, एकवचनम्
गम्-धातुः, शतृ, नपु., सप्तमी, एकवचनम्

शब्दत्रयेन शाब्दबोधत्रयं शक्यं पाणिनि-व्याकरणानुसारम् , न तु त्रयाधिकम् एकसंज्ञात्वात् [आ कडारादेका संज्ञा (1/4/1)] प्रतिशब्दमेक एव संबन्धः प्राप्नोति।

रामः वनं गच्छति।

1. वनकर्मकरामकर्तृकगमनानुकूलो व्यापारः।
2. गमनसमानकालिको वनकर्मकः दानानुकूलो व्यापारः।
3. वनकर्मकगमनसमानकालिकः दानानुकूलो व्यापारः।

---

6 आकाङ्क्षा योग्यता संनिधिश्च वाक्यार्थज्ञानहेतुः। (तर्कसंङ्ग्रहे)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

अत्र शाब्दबोधवाक्यत्रये द्वितीयमसाध्वपूर्णत्वादाकाङ्क्षायास्ततोऽगमकत्वात्। प्रथमतृतीययोः साधुत्वं सिद्धं व्याकरणेन। द्वयोः साधुत्वे सत्यपि सङ्गणकं प्रथमं प्रथमस्य साधुत्वमङ्गीकरोति। यतोहि तृतीये क्रियाद्वयस्य सर्वा आकाङ्क्षा न पूर्यन्ते, परं प्रथमवाक्ये गच्छतेः कर्तृकर्मणोः आकाङ्क्षा पूर्यते। तस्मात् प्रथमस्य साधुत्वं साध्यते सङ्गणकेन। इत्थमाकाङ्क्षां आश्रित्य शाब्दबोधे साहाय्यं क्रियते सङ्गणकेन।

योग्यता –

पूर्वमेवोक्तं यद्योग्यतायाः प्रयोगोऽस्माभिरधुना न विधीयते, प्रयोगे काठिन्यात्। परं योग्यतायाः प्रयोगः कथं कार्यते तत् पश्याम तावत्।

'शकटं वनं गच्छति' इत्येतस्मिन् वाक्ये प्रतिशब्दमेकाधिकं विश्लेषणम्।

यथा –

शकटम् – नपुंसकम्, प्रथमा, एकवचनम्
नपुंसकम्, द्वितीया, एकवचनम्

वनम् – नपुंसकम्, प्रथमा, एकवचनम्
नपुंसकम्, द्वितीया, एकवचनम्

गच्छति – गम्-धातुः, लट्, कर्तरि, प्र.पु., एकवचनम्
गम्-धातुः, शतृ-प्रत्ययः, पुल्लिङ्गम्, सप्तमी, एकवचनम्
गम्-धातुः, शतृ-प्रत्ययः, नपुंसकम्, सप्तमी, एकवचनम्

अत्राप्येकसंज्ञात्वात् षड् शाब्दबोधावाक्यानि प्राप्यन्ते –

शकटं वनं गच्छति।

1. शकटकर्तृको वनकर्मको गमनानुकूलो व्यापारः।
2. वनकर्तृको शकटकर्मको गमनानुकूलो व्यापारः।
3. शकटाभिन्नवनकर्तृको गमनानुकूलो व्यापारः।
4. शकटाभिन्नवनकर्मको गमनानुकूलो व्यापारः।
5. वनाभिन्नशकटकर्तृको गमनानुकूलो व्यापारः।
6. वनाभिन्नशकटकर्मको गमनानुकूलो व्यापारः।

वन-शब्दे गमनानुकूलव्यापाराश्रयत्वरूपयोग्यताभावादसाधुत्वं द्वितीयशाब्दबोधवाक्यस्य। शकटे च संज्ञात्वाद् विशेषणयोग्यताभावादसाधुत्वं तृतीयतुरीययोः। तथा च वनेऽपि, तस्मादसाधुत्वं पञ्चमषष्ठयोः। इत्थं योग्यताबलेन केवलं प्रथमस्य साधुत्वम्। परं कठिनः सङ्गणकेन योग्यतायाः प्रयोगः। योग्यतानादरेण सर्वेषां साधुत्वे सत्यपि सङ्गणकं प्रथमं प्रथमद्वितीययोः साधुत्वमङ्गीकरोत्याकाङ्क्षायाः पूर्णत्वात्।

सन्निधिः –

'रामः दुग्धं पीत्वा शालां गच्छति' इत्येतस्मिन् वाक्ये नैकेषु शब्देष्वेकाधिकं विश्लेषणं प्राप्नोति।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

यथा -

रामः - पुल्लिङ्गम्, प्रथमा, एकवचनम्

रा-धातुः, लट्, कर्तरि, उ.पु., बहुवचनम्

दुग्धम् - नपुंसकम्, प्रथमा, एकवचनम्

नपुंसकम्, द्वितीया, एकवचनम्

दुह्-धातुः, क्त-प्रत्ययः, पुल्लिङ्गम्, द्वितीया, एकवचनम्

दुह्-धातुः, क्त-प्रत्ययः, नपुंसकम्, प्रथमा, एकवचनम्

दुह्-धातुः, क्त-प्रत्ययः, नपुंसकम्, द्वितीया, एकवचनम्

दुह्-धातुः, परस्मैपदम्, कर्तरि, लोट्, म.पु., द्विवचनम्

पीत्वा - पा-धातुः, क्त्वा-प्रत्ययः

शालाम् - स्त्रीलिङ्गम्, द्वितीया, एकवचनम्

गच्छति - गम्-धातुः, लट्, कर्तरि, प्र.पु., एकवचनम्

गम्-धातुः, शतृ-प्रत्ययः, पुल्लिङ्गम्, सप्तमीं, एकवचनम्

गम्-धातुः, शतृ-प्रत्ययः, नपुंसकम्, सप्तमीं, एकवचनम्

शब्दपञ्चकात् शाब्दबोधवाक्यद्वयं प्राप्यते एकसंज्ञात्वात्

रामः दुग्धं पीत्वा शालां गच्छति।

1. दुग्धकर्मकपानोत्तरकालिकः शालाकर्मको रामकर्तृको गमनानुकूलव्यापारो वर्ण्यतेऽस्मिन्वाक्ये।

अथवा

2. शालाकर्मकपानोत्तरकालिको दुग्धकर्मकरामकर्तृकगमनानुकूलव्यापारो वर्ण्यतेऽस्मिन्वाक्ये।

काव्यात्मकभाषां विहाय एकतिङ्वाक्ये वाक्यादौ कर्तुः कर्मणश्च क्रियातः पूर्वं स्थानं निश्चित्य संन्निधेः प्रयोगः क्रियतेऽत्र। सन्निधिगणनार्थमादितः प्रतिशब्दमङ्क्यते। सम्बन्धितशब्दयोर्मध्येऽन्तरालं निस्सार्य आहत्य च योऽङ्कः प्राप्यते सोऽत्र सन्निधिरिति। यथा -

प्रथमवाक्ये तावत् सन्निधिं पश्याम -

रामः	दुग्धं	पीत्वा	शालां	गच्छति
1	2	3	4	5

रामः = गच्छतेः कर्त्ता, तस्मात् 5 (गच्छतेः स्थानम्) - 1 (रामशब्दस्य स्थानम्) = 4

दुग्धम् = पिबतेः कर्म, तस्मात् 3 (पिबतेः स्थानम्) - 2 (दुग्धशब्दस्य स्थानम्) = 1

पीत्वा = गच्छतेः पूर्वकालिकक्रिया, तस्मात् 5 (गच्छतेः स्थानम्) - 3 (पिबतेः स्थानम्) = 2

शालाम् = गच्छतेः कर्म, तस्मात् 5 (गच्छतेः स्थानम्) - 4 (शालाशब्दस्य स्थानम्) = 1

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

आहत्य, 4 + 1 + 2 + 1 = 8

द्वितीयवाक्ये सन्निधिं पश्यामः –

रामः दुग्धं पीत्वा शालां गच्छति
1 2 3 4 5

रामः = गच्छतेः कर्ता, तस्मात् 5 (गच्छतेः स्थानम्) – 1 (रामशब्दस्य स्थानम्) = 4

दुग्धम् = गच्छतेः कर्म, तस्मात् 5 (गच्छतेः स्थानम्) – 2 (दुग्धशब्दस्य स्थानम्) = 3

पीत्वा = गच्छतेः पूर्वकालिकक्रिया, तस्मात् 5 (गच्छतेः स्थानम्) – 3 (पिबतेः स्थानम्) = 2

शालाम् = पिबतेः कर्म, तस्मात् 4 (शालाशब्दस्य स्थानम्) – 3 (पिबतेः स्थानम्) = 1

आहत्य, 4 + 3 + 2 + 1 = 10

वाक्येषु यावदधिकसन्निधिवाक्यं सङ्गणकेन प्रथममङ्गीक्रियते। अत्र वाक्यद्वये शालाकर्मकगमनानुकूलव्यापारयुक्तं वाक्यं सान्निध्याधिक्यात् सङ्गणकेन साधु मन्यते। अन्यभाषावत् संस्कृते शब्दाः स्थानदृष्ट्या वाक्ये न नियुक्ताः। अतोऽन्यान्यपि वाक्यानि शब्दपञ्चकाद् भवितुमर्हन्ति। यथा –

2. रामः पीत्वा शालां दुग्धं गच्छति।

3. रामः दुग्धं शालां पीत्वा गच्छति।

भाषायां कारकक्रिययोर्मध्ये क्रियान्तरस्य कारकं न प्रयुज्यतेऽगमकत्वात्। अत इदं वाक्यद्वयं तु सर्वथैवाग्राह्यम्। यतोहि द्वितीयवाक्ये शालाकर्मको गच्छतिरतः 'शालां' 'गच्छति' इत्येतयोर्मध्ये 'दुग्धं' इति शब्दो न स्यात्, पिबतेः कारकत्वात्। तृतीये च 'शालां' इति गच्छतेः कारकत्वाद् 'दुग्धं' 'पीत्वा' इत्येतयोर्मध्ये 'शालाम्' इत्येतस्योपस्थानमसाधुरेव, दुग्धकर्मकत्वात् पिबतेः।

"बालकः हस्तेन कदलीफलं छित्वा खादन्तं वानरं पश्यति" – अस्मिन् वाक्ये 'हस्तेन' 'कदलीफलम्' 'छित्वा' 'खादन्तम्' इत्यादीनि न बालककर्तृकपश्यतेः कारकाणि। पुनरपि 'बालकः' 'पश्यति' इत्येतयोर्द्वयोर्मध्ये तेषामुपस्थितिः साधुरेव। यतोह्येतानि सर्वाणि पश्यतेः कर्मकारकस्य वानरस्य विशेषणानि, विशेषणत्वात् साधुत्वम्।

4. रामः शालां पीत्वा दुग्धं गच्छति।

गमकत्वे सत्यपीदं वाक्यं सामान्यरूपेण न स्वीक्रियन्ते। वाक्यादौ कर्तुः क्रियातः पूर्वं कर्मणश्च स्थानं निश्चित्य यदि सन्निधेः प्रयोगो विधीयते चेत् सम्प्रति सङ्गणकं शाब्दबोधे साहाय्यं प्रतिजानीते। इत्थं सन्निधेर्माहात्म्यं शाब्दबोधाय सङ्गणकेन स्वीक्रियते।

गाणितिक-संरचना –

गाणितिककार्यार्थं वाक्यस्थशब्दा द्विधा विभज्यन्ते। 1) आकाङ्क्षकाः = येषामाकाङ्क्षा

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

वर्तते। 2) आकाङ्क्ष्याः = ये आकाङ्क्षकानाम् आकाङ्क्षां पूरयन्ति। प्रत्येकमपि शब्दानां वाक्ये यः स्थानक्रमांकः सः "node" मध्ये स्थाप्यते। तथा प्रतिशब्दं विश्लेषितरूपस्य या सङ्ख्या सा 'cell' मध्ये स्थाप्यते। [i, j] अर्थात् वाक्ये 'i' स्थाने वर्तमानस्य शब्दस्य j'th विश्लेषकरूपम् j'thcell मध्ये स्थितम्। उदाह्रियतेऽत्र 'रामः वनं गच्छति' अस्मिन् वाक्ये 'रामः' इति प्रथमस्थाने वर्तते। अतः 'node' मध्ये '1' इति संरक्ष्यते। 'रामः' इत्यतस्य द्वे विश्लेषितरूपे ये 'cell' मध्ये स्थाप्येते -

1) राम-प्रातिपदिकम्, पुल्लिङ्गम्, प्रथमा, एकवचनम्

2) रा-धातुः, लट्, कर्तरि, उ. पु., बहुवचनम्।

यदि 'रामः' सुबन्तं पदं तर्हि i=1, j=1 अर्थात् 'रामः = [1, 1]', यदि रामः इति तिङन्तं पदं तर्हि i=1, j=2 अर्थात् 'रामः = [1, 2]' इति।

गच्छतेरपि त्रीणि विश्लेषित-रूपाणि -

1) गम्-धातुः, लट्, कर्तरि, प्र. पु., एकवचनम्,

2) गम्-धातुः, शतृ-प्रत्ययः, पुल्लिङ्गम्, सप्तमी, एकवचनम्,

3) गम्-धातुः, शतृ-प्रत्ययः, नपुंसकम्, सप्तमी, एकवचनम्।

प्रदत्तवाक्ये 'गच्छति' तिङन्तं वर्तते अतः 'गच्छति=[3, 1]'।

प्रथमसोपानम् : सोपानेऽस्मिन् सर्वेषां पदानां विभक्तिं दृष्ट्वा द्वयोः पदयोः मध्ये स्थितान् सम्बन्धाननुशील्य विश्लेषके दर्शयामः। अधुना '3Dimensional matrix' मध्ये 'रामः' 'गच्छति' इति द्वयोर्मध्ये संबन्धं दर्श्यते सङ्गणकेन - ([1, 2] R [3, 1]) इति। अत्र प्रथमं युगलं [1, 2] इति आकाङ्क्ष्यशब्दस्य वर्तते द्वितीयञ्च [3, 1] इति आकाङ्क्षकस्य। द्वयोर्मध्ये यः संबन्धः सः 'R' इति निर्दिश्यते। केचन संबन्धनिश्चायका नियमा लिखिताः, यैः सङ्गणकस्य संबन्धदर्शने सामर्थ्यं वर्तते। संक्षिप्तरूपेण ते नियमा अत्रोदाह्रियन्ते -

नियमः 1 , यदि वाक्ये

- कश्चन शब्दः 'w' इति प्रथमाविभक्तौ, एकवचने
- काचित् क्रिया 'v' इति कर्तरिवाच्ये, एकवचने, प्रथमपुरुषे

तर्हि 'w' शब्दः 'v' क्रियायाः कर्त्ता संभवति।

नियमः 2, यदि वाक्ये

- कश्चन शब्दः 'w' इति द्वितीयाविभक्तौ, एकवचने
- काचित् क्रिया 'v' इति कर्तरिवाच्ये, सकर्मका

तर्हि 'w' शब्दः 'v' क्रियायाः कर्म संभवति।

नियमः 3, यदि वाक्ये

- कश्चन शब्दः 'w' इति सप्तमीविभक्तौ
- काचित् क्रिया 'v' इति

तर्हि 'w' शब्दः 'v' क्रियायाः अधिकरणं संभवति।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

अधुना चिन्त्यतां वाक्यमेकम्, यथा – रामः वनं गच्छति।

1 2 3 वाक्ये

विद्यमानशब्दानां स्थानक्रमांकः

रामः –

1) राम-प्रातिपदिकम्, पुल्लिङ्गम्, प्रथमा, एकवचनम् = रामः [1, 1]

2) रा-धातुः, लट्, कर्तरि, उ. पु., बहुवचनम् = रामः [1, 2]

वनम् –

1) वन-प्रातिपदिकम्, नपुंसकलिङ्गम्, प्रथमा, एकवचनम् = वनम् [2, 1]

2) वन-प्रातिपदिकम्, नपुंसकलिङ्गम्, द्वितीया, एकवचनम् = वनम् [2, 2]

गच्छति –

1) गम्-धातुः, लट्, कर्तरि, प्र. पु., एकवचनम् = गच्छति [3, 1]

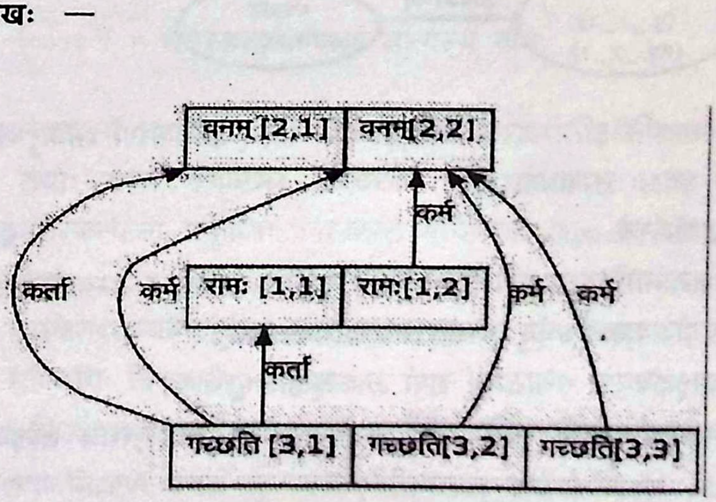
2) गम्-धातुः, शतृ-प्रत्ययः, पुल्लिङ्गम्, सप्तमी, एकवचनम् = गच्छति [3, 2]

3) गम्-धातुः, शतृ-प्रत्ययः, नपुंसकलिङ्गग्, सप्तमी, एकवचनम् = गच्छति [3, 3]

इदं स्रोतसामग्रीरूपेण गृहीत्वा नियगानां साहाय्येन सङ्गणकं संभावितानि परिणामानि इत्थं दर्शयति –

आकाङ्क्य-शब्दाः आकाङ्क्षक-शब्दाः	संभावित-संबन्धाः	
[2, 2]	कर्म	[3, 2]
[2, 2]	कर्म	[3, 3]
[2, 2]	कर्म	[1, 2]
[2, 2]	कर्म	[3, 1]
[2, 1]	कर्ता	[3, 1]
[1, 1]	कर्ता	[3, 1]

परिणमित-आलेखः —



CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

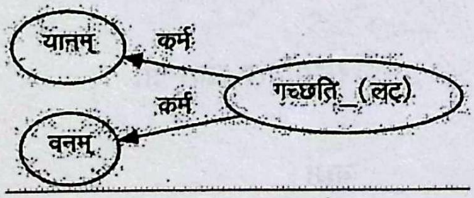
उपर्युक्तचित्रेणास्माभिः वाक्येविद्यमानां यथासम्भवसम्बन्धानां प्रदर्शनं क्रियते। यदि तत्र एकस्मिन् समये द्वौ शब्दौ पश्यामः तर्हि समस्ताः सम्बन्धाः सम्भाविताः। परन्तु यदा एतैः शब्दैः किमपि सार्थकं वाक्यं भवति तर्हि एते समस्तसम्बन्धाः असम्भाविताः भवन्ति। किञ्च अन्यशब्दानाम् उपस्थित्या तत्र प्रतिबन्धाः उपस्थापिताः भवन्ति यस्मात् कस्मिन्नपि वाक्ये यदि n शब्दः अस्ति चेत् केवलं n-1 सम्बन्धः एव व्यक्तः भवति।

द्वितीयसोपानम् : समस्तसम्भावितसम्बन्धात् केवलं n-1 सम्बन्धानां चयनं करणीयं येन वाक्ये उपस्थितानां सर्वे सम्बन्धानां मध्ये सम्बन्धः प्रस्थापितः भवतु।

(अ) अस्मादालेखाद् अपेक्षितसंबन्धालेखाय (Tree) सीमाङ्कनं क्रियते -

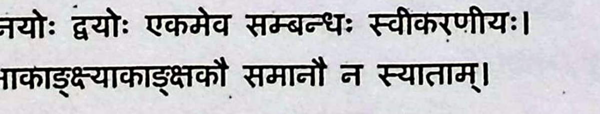
1) आकाङ्क्ष्यशब्द एकामेव आकाङ्क्षां पूरयेत्।

2) आकाङ्क्षकस्य आकाङ्क्षा एकेनैव शब्देन पूरयेत्। एकाधिकशब्दे प्राप्ते सान्निधिमनुसरेत्।



चित्रेऽस्मिन् गच्छति इति पदस्य कर्माकाङ्क्षां यानम् एवञ्च वनम् द्वावपि शब्दौ पूरयतः परन्तु अनयोः द्वयोः एकमेव सम्बन्धः स्वीकरणीयः।

3) आकाङ्क्ष्याकाङ्क्षकौ समानौ न स्याताम्।



प्रकृतस्थले गच्छति इति पदस्य गच्छति इति पदम् अधिकरणं भवितुं नार्हति।

4) सर्वे शब्दाः सम्बध्येत्।

5) शब्दयोर्मध्ये - एकाधिकसंबन्धश्चेद् विशेषणकर्मसमानाधिकरणकर्त्तासमानाधिकरणाधिकरणापादान-सम्प्रदानकरण-कर्मप्रयोज्यप्रयोजककर्त्ताकर्त्तृषु उत्तरोत्तरस्य बलीयस्त्वम्।

6) आकाङ्क्षकस्य यथासंभवं सर्वा आकाङ्क्षाः पूरयेयुः।

=> सीमाङ्कनस्योल्लंघने सति अपेक्षितसंबन्धालेखो यथोल्लंघनं दण्ड्यते। दण्डनात् न

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

तस्य प्राधान्यम्।

कथं प्राप्यते अपेक्षितसंबन्धालेखः सीमाङ्कनात्तत् पश्यतु तावत् –

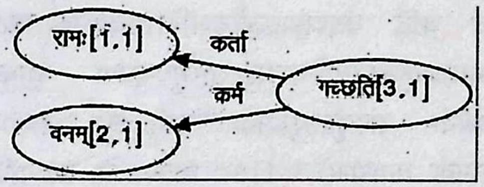
1) यदि [2, 2]-शब्दः [3, 2]-शब्दस्य कर्म तर्हि [1, 2]-शब्दस्य कर्माकाङ्क्षा तथा [3, 2]-शब्दस्य कर्त्राकाङ्क्षा न पूर्येत्।

2) यदि [2, 2]-शब्दः [3, 3]-शब्दस्य कर्म तर्हि उपरिवदत्रापि।

3) यदि [2, 2]-शब्दः [1, 2]-शब्दस्य कर्म तर्हि 'गच्छति' शब्दः कुत्रापि न युज्यते।

4) यदि [2, 1]-शब्दः [3, 1]-शब्दस्य कर्त्ता तर्हि 'रामः' शब्दोऽवशिष्यते। अतो उपरिवत् अयमपि दण्ड्यते।

यदि [2, 2]-शब्दः [3, 1]-शब्दस्य कर्म, [1, 1]-शब्दः [3, 1]-शब्दस्य कर्त्ता भवति चेत् सर्वसीमाङ्कनस्य परिपालनादस्यापेक्षितसंबन्धालेखस्य प्राधान्यम्।



(ब) द्वितीयचरणस्य प्रतिबन्धानाम् तृतीयचरणे उपयुज्यमानानन्तरं वाक्यविश्लेषणं लभ्यते। सामान्यतः एते एकाधिकाः भवन्ति। उदाहरणं पश्यामः

यानं वनं गच्छति।

पदज्ञानाधारेण वाक्येऽस्मिन् यानम् एवञ्च वनम् द्वावपि कर्ता एवं कर्म भवितुमर्हतः। अस्मात् कारणात् द्विधा वाक्यविश्लेषणं प्राप्यते। परन्तु एतयोः किं वाक्यविश्लेषणं सम्यगस्ति अस्य निर्धारणं कथं भवति ? अस्य निर्धारणार्थं वाक्ये स्थितानां शब्दानां मध्ये यत् स्थानं तेन कृतम्। सहैव सम्बन्धानां पौर्वापर्यमपि निश्चितम्। उदाहरणार्थम् – यदा किमपि पदं तृतीयाविभक्त्या हेतुः अथवा करणं भवितुमर्हति एवं क्रियापदं करणस्याकाङ्क्षा वर्तते। तदा हेतोरपेक्षा करणस्य प्राधान्यं दातव्यम्। आभ्यां प्रतिबन्धाभ्यां शब्दविश्लेषणस्य मूल्यं निम्नरीत्या निर्धारितम् –

मूल्यम् (cost) = समस्तसम्बन्धानां मूल्यस्य योगः

प्रत्येकसम्बन्धस्य मूल्यम् = सम्बन्धस्य श्रेणी x ययोः द्वयोः सम्बन्धस्य मध्ये स्थितः यः सम्बन्धः, तयोः मध्ये स्थानम्। सम्बन्धः यदि कारकेतरः तर्हि तस्य मूल्यम् 100 निर्धारितम्। कारकसम्बन्धानां अनुक्रमः पत्रस्यान्ते सूचिकायां (Appendix) दत्तः।

मूल्यस्यास्य कारणम् –

अ) कारकेतरसम्बन्धापेक्षया कारकसम्बन्धस्यैव प्राधान्यम्।

ब) सन्निधिः विचारार्थं स्वीकृता।

प्रायोगिकदृष्ट्या निर्माणम् :

वाक्यविश्लेषणनिर्माणे प्रथमं तु व्याकरणनियमान् दृष्ट्वा वाक्यं स्थापितम्। अग्रिमस्तरे

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

गाणितिकप्रक्रियामुपयुज्य वाक्यविश्लेषणानां सङ्कलनं भवति। सङ्कलनान्ते विप्रतिषेधः समायाति। विप्रतिषेधसमाधानेन (Constraint Solver) तेषां सर्वेषां वाक्यानां श्रेणीकरणम्। ततः श्रेण्याः प्रथमस्तरवाक्यविश्लेणं प्रस्तौति। एतदर्थं CLIPS एवञ्च MINION इति द्वे तन्त्रांशे उपयुज्येते। 'वाक्यविश्लेषकम्' अन्तर्जाले (Internet) अधोलिखिते सङ्केते विद्यते -

http://sanskrit.uohyd.ernet.in/~anusaaraka/sanskrit/MT/test_skt.html

CLIPS एवञ्च MINION इति द्वयमाधारीकृत्य एतत् 'वाक्यविश्लेषकम्' निर्मितम्। तस्य कारणं एते द्वे General Public License मध्ये उपस्थिते, तस्मात् वयं एतयोः उपयोगं कुर्मः। एवमेताभ्यां सह अस्माकं परिचयोऽप्यस्ति। वस्तुतः सैद्धान्तिकरूपेण वाक्यविश्लेषणनिर्माणार्थमेते उपयुज्येते। अन्यतन्त्राशमुपयुज्यैतत् कार्यं कर्तुं शक्यते। तत्र नास्ति कश्चनावरोधः।

कार्यसम्पादनस्तरः

वर्तमानवाक्यविश्लेषकयन्त्रं यदि पदच्छेदान्वयसहितएकतिङ्गद्यात्मकं वाक्यमस्ति चेत् सम्यक्कार्यं सम्पादयति। मानवकृतानुबन्धितसङ्ग्रहमुपयुज्य वाक्यविश्लेषकस्य परीक्षणं कृतमस्माभिः। प्राथमिकस्तरीयमेकं संस्कृतपुस्तकं स्वीकृत्य ततः उपर्युक्तप्रतिज्ञानुसारेण (पदच्छेदान्वयसहितएकतिङ्गद्यात्मकं वाक्यम्) 113 वाक्यानि सङ्गृहीतानि। एतानि वाक्यानि सन्दर्भानुसारं मानवकृतानुबन्धितानि आसन्। एते 525 शब्दात्मकाः आसन्। प्रत्येकं वाक्यम् न्यूनतमम् द्विशब्दात्मकम् आसीत् एवञ्च अधिकतमं चतुर्दशशब्दात्मकम् इति। शब्दविश्लेषकः (Morphological Analyser) वाक्यविश्लेषकस्य मूलभूतावश्यकता। सर्वप्रथमं शब्दविश्लेषकेन ये शब्दाः विश्लेषिताः न भवन्ति तेषां विश्लेषणं यन्त्राय दद्मः। अनन्तरमग्रिमं तन्त्रांशं चालयामः। इदानीं वाक्यविश्लेषकः एकाधिकं वाक्यविश्लेषणं ददाति। परीक्षणार्थः प्रथमं वाक्यविश्लेषणं स्वीक्रियते। तेषु 113 वाक्येषु 97 (86%) वाक्यानि सम्यग्विश्लेषितानि वाक्यविश्लेषणेन। एवञ्च 16 (14%) वाक्यानां विश्लेषणेषु एकमशुद्धमासीत्। तत्र 10 वाक्येषु सम्बन्धस्य नाम अशुद्धमासीत्। 3 वाक्येषु सम्बन्धस्य योजने अशुद्धिः एवं 3 वाक्येषु सम्बन्धस्य नाम एवं योजनं द्वयमपि अशुद्धमासीत्।

उपर्युक्तप्रक्रियायां यत्र वाक्यविश्लेषकेन सप्तमीविभक्तिः दृष्टा तत्र तेन अधिकरणं सम्बन्धस्य नाम दत्तम् परन्तु मानवकृतानुबन्धसङ्ग्रहे कालाधिकरणम् अथवा देशाधिकरणम् इति सम्बन्धस्य नाम आसीत्, अनेनैव तृतीयां दृष्ट्वा हेतुः करणम् इत्येतयोः मध्ये सम्बन्धनामभेदः आसीत्, पुनश्च द्विकर्मकधातोः किम् गौणकर्म एवञ्च किं मुख्यकर्म अत्रापि अशुद्धिः प्रदत्ता। चुरादिगणपठितानां धातूनां रूपाणि णिजन्तेऽपि समानानि भवन्ति। अतः तत्र स्वार्थणिच् स्वीकृत्य कर्ता एवं कर्म इति सम्बन्धनाम योजनीयम् उत णिजन्तं स्वीकृत्य प्रयोज्यकर्ता एवं प्रयोजककर्ता इति स्वीकरणीयम्। अनेन प्रकारेण अशुद्धविश्लेषणानां विवेचनं प्राप्यते।

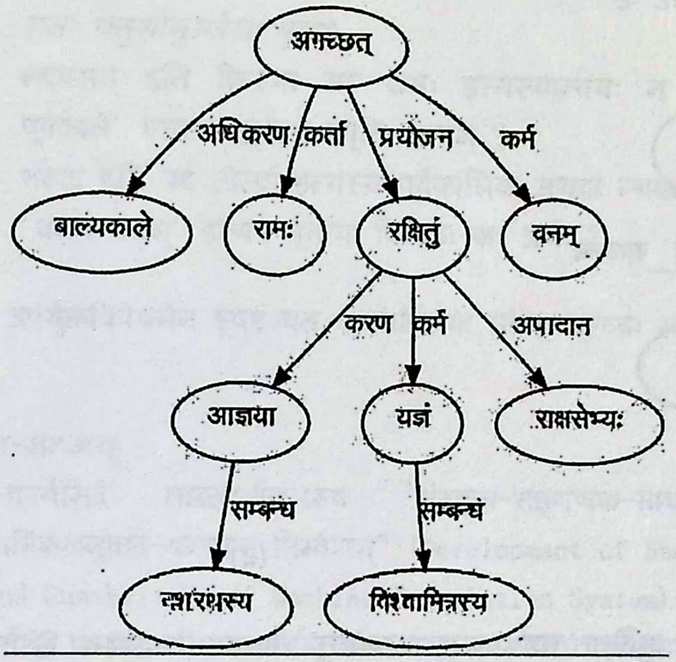
वाक्यस्यैकस्य विश्लेषणं पश्यामः -

बाल्यकाले रामः दशरथस्य आज्ञया विश्वामित्रस्य यज्ञं राक्षसेभ्यः रक्षितुं वनम् अगच्छत्।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

वाक्यस्यास्य विश्लेषणमनेन प्रकारेण प्रदर्शयति वाक्यविश्लेषकम् --



वाक्यविश्लेषकनिर्माणे समस्याः

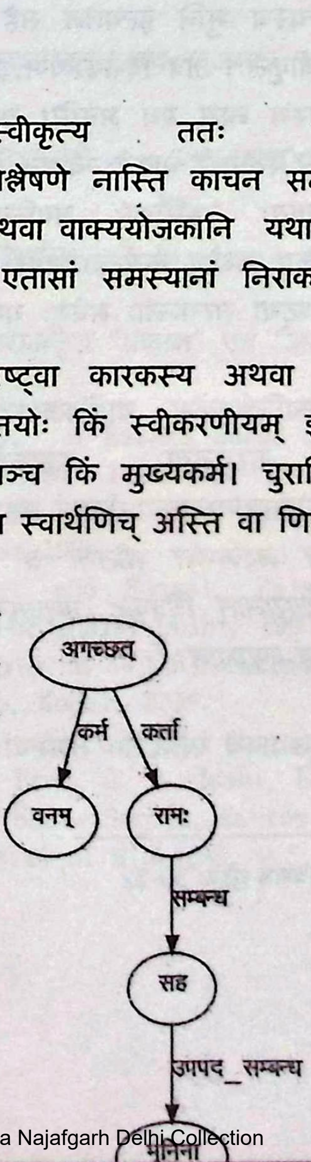
प्राथमिकस्तरीयमेकं संस्कृतपुस्तकं स्वीकृत्य ततः प्रतिज्ञानुसारेण (पदच्छेदान्वयसहितएकतिङ्गद्यात्मकं वाक्यम्) वाक्यविश्लेषणे नास्ति काचन समस्या। परन्तु तत्रैव यदि वाक्ये द्वे क्रिये अथवा बह्वीः क्रियाः वर्तन्ते अथवा वाक्ययोजकानि यथा - यदा-तदा, यथा-तथा, अथ, तस्मात्, इत्यादयः आगच्छन्ति। एतासां समस्यानां निराकरणमग्रिमस्तरे करिष्यते।

अनेन प्रकारेण द्वितीया समस्या विभक्तिं दृष्ट्वा कारकस्य अथवा उपपदविभक्तेः निर्धारणम्। तृतीयाविभक्तिं दृष्ट्वा हेतुः करणम् इत्येतयोः किं स्वीकरणीयम् इत्यस्ति एका समस्या। अन्या च द्विकर्मकधातोः किम् गौणकर्म एवञ्च किं मुख्यकर्म। चुरादिगणपठितानां धातूनां रूपाणि णिजन्तेऽपि समानानि भवन्ति। अतः तत्र स्वार्थणिच् अस्ति वा णिजन्त इत्यपरा समस्या।

तृतीया तु उपपदेषु समस्या वर्तते। उपपदानि द्योतकानि वाचकानि वा ? अत्र वयं_ उपपदानां वाचकत्वमङ्गीकुर्मः। वाचकत्वे प्रमाणमग्रिमोदाहरणेन प्रतिपादयामः -

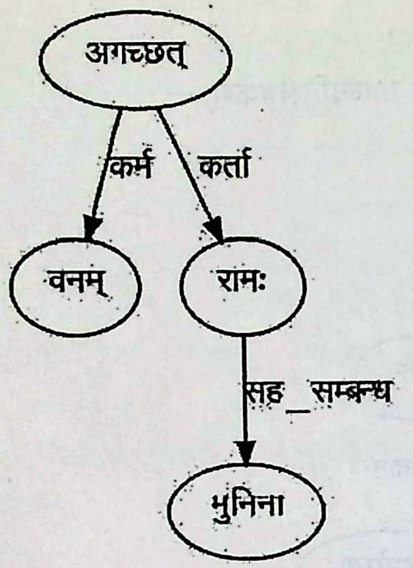
रामः मुनिना सह वनम् अगच्छत्।

एतस्य विश्लेषणं चित्रे पश्यामः -



CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.



चित्रम् : (अ) (ब)

उपर्युक्तचित्रयोः 'रामः मुनिना सह वनम् अगच्छत्' एतस्य वाक्यस्य द्विधा विश्लेषणं निरूपितम्। चित्रे (अ) इत्यत्र राम एवं मुनि इत्येतयोः मध्ये सह (उपपदम्) इति सम्बन्धत्वेन अङ्गीकृतम् तदेव तस्य द्योतकत्वं द्योतयति। परन्तु चित्रे (ब) राम इत्यस्यान्वययः सह इत्यनेन एवं सह इत्यस्य मुनि इत्यनेन सहे भवति। तदेव तस्य वाचकत्वं सूचयति। अयमेव पक्षः अस्माभिः स्वीकृतः। अत्र विशेषरूपेण द्रष्टव्यम् यदि वयं सह इत्यस्य द्योतकत्वं स्वीकुर्मश्चेत् तर्हि तद् सम्बन्धस्य नाम एव भवति। वाक्ये तस्यान्वययः कथम्। न कर्तुं शक्यते। अतः तस्य वाचकत्वमिति स्वीकारे वाक्येऽन्वयात् दोषः नोपपद्यते।

विभक्तयः बह्वर्थिकाः भवन्ति इति। उदाहरणार्थं द्वितीयाविभक्तिः क्रियाविशेषणे, कालाध्ववाचकेषु शब्देषु कर्मत्वाद्भवति। तस्य निर्धारणार्थं योग्यतायाः ज्ञानमावश्यकम्। अस्माकं कोशाः अर्थदृष्ट्या सम्पन्नाः भवेयुः यथा के क्रियाविशेषणानि के कालवाचका के च मार्गवाचकाः इत्यादि।

'वाक्यविश्लेषकम्' यान्त्रिकदृष्ट्या कार्यं करोति। यत्र नियमानामुल्लङ्घनं भवति तानि स्थलानि सरलतया दृश्यन्ते। यथा अधोलिखितमुदाहरणं पश्यामः। सामान्यरूपेणोदाहरणस्यास्यार्थज्ञानं सहजतया भवति। परन्तु तत्र का समस्या ? इति।

गुहेन लक्ष्मणेन सीतया च सहितः रामः वनेन वनं गत्वा बहुदकाः नदीः तीर्त्वा भरद्वाजस्य शासनात् चित्रकूटं अनुप्राप्य वने रम्यं आवसथं कृत्वा देवगन्धर्वसङ्काशाः ते त्रयः रममाणः सुखं न्यवसन्। [7]

उपर्युक्तवाक्ये एतादृशाः समस्याः दृश्यन्ते :

7 सङ्क्षेपरामायणम् श्लोकः 30-32

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

(अ) 'ते त्रयः' इति पदद्वयं कं बोधयति ? (वाक्ये तु *गुहेन लक्ष्मणेन सीतया च सहितः रामः चतुर्णामुल्लेखः कृतः*)

(ब) न्यवसन् इति क्रियया सह रामः इत्यस्यान्वयः न भवति। अतः 'समानकर्तृकयोः पूर्वकाले' एतस्य सूत्रस्य प्रवृत्तिः कथम् ?

(स) गत्वा इति पदं तीर्त्वा इत्यस्य पूर्वकालिकं अथवा न्यवसन् इत्यस्य ?

(द) 'वनेन वनम्' इत्यत्र तृतीया विभक्तेः कः अर्थः ?

उपर्युक्तविवेचनेन स्पष्टं यत् वाक्यविश्लेके सन्ति बहव्यः समस्याः। तथापि स उपयुक्तः।

कृतज्ञता-ज्ञापनम्

कार्यमिदं भारतसर्वकारस्य "संस्कृत-सङ्गणक-साधनानि एवं संस्कृत-हिन्दी-यान्त्रिकानुवाद-यन्त्रस्य निर्माणम्" (Development of Sanskrit Computational tools and Sanskrit-Hindi Machine Translation System) इति नाम्नी परियोजनायाः भागं वर्तते। परमादरणीयानाचार्यान् प्रो० रामकृष्णमाचार्यमहाभागान् सविनयं कार्तज्ञं विज्ञापयामः। येषां मार्गदर्शनं सदा सर्वदा प्राप्यते।

सन्दर्भग्रन्थसूची

1. कुटुम्बशास्त्री वी०, सङ्क्षेपरामायणम्, संस्कृत-स्वाध्याय-योजना, राष्ट्रिय-संस्कृत-संस्थानम्, नई दिल्ली

2. पाण्डेय, गोपालदत्त, वैयाकरणसिद्धान्तकौमुदी, (मूलमात्रम्) पुनर्मुद्रित, चौखम्भाविद्यावन वाराणसी, 2000.

3. रामानुजताताचार्य, एन.एस., शाब्दबोधमीमांसा, Institute Francis De Pondicherry, 2005.

4. शर्मा, रघुनाथ, वाक्यपदीयम् 3 भागः, हेलाराजकृत "प्रकाश" एवं "अम्बकर्त्री" टीका सहित, वाराणसेय संस्कृतविश्वविद्यालय वाराणसी, 1974.

5. Bharati, Akshar and Sangal, Rajeev, A Karaka Based Approach to Parsing of Indian Languages. COLING90: Proc. of Int. Conf. on Computational Linguistics (Vo 1. 3), Helsinki}, Association for Computational Linguistics, NY, August 1990.

6. Bharati, Akshar, Chaitanya, Vineet and Sangal, Rajeev, NLP A Paninian Perspective, Prentice Hall of India, Delhi, 1994.

7. Cardona George, Pā▪ini and pā▪iniyas on śe▪a Relations, Kunjunni Raja Academy of Indological Research, Kochi, 2007.

8. Dash Achyutanand, The syntactic role of adhi in the Pāninian Kāraka system in Pā▪iniyan Studies Prof. S. D. Joshi, Felicitation volume, ed. Deshpande Madhav M, and Bhate Saroja, Center for South and Southeast Asian Studies, University of Michigan, U.S.A., 1991.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

9. Gent, Ian P., Jefferson, Chris and Miguel, Ian. MINION: A Fast, Scalable, Constraint Solver, The European Conference on Artificial Intelligence 2006 (ECAI 06).}
10. Gillon Brendan S. Word Order in Classical Sanskrit. Indian Linguistics, v.57, n.1, pp. 1-35, 1996.
11. Gillon Brendan S. Bhart▪haii's rule for unexpressed kārakas: The problem of control in Classical Sanskrit. Indian Linguistic Studies, Festschrift in Honor of George Cardona, Ed. Deshpande, Hook, Motilal Banarasidass, Delhi, 2002.
12. Hellwig, Oliver, Extracting Dependency Trees from the Sanskrit Texts, Proceedings of the Sanskrit Computational Linguistics Symposium, Ed. Kulkarni and Huet, LNAI 5406, Springer Verlag, 2009.
13. Huet, Gérard, Formal Structure of Sanskrit Text: Requirements Analysis for a Mechanical Sanskrit Processor, Proceedings of the Sanskrit Computational Linguistics Symposium, Ed. Huet, Kulkarni and Sharf, LNAI 5402, Springer Verlag, 2009.
14. Huet, Gérard, Shallow syntax analysis in Sanskrit guided by semantic nets constraints, Proceedings of International Workshop on Research Issues in Digital Libraries, Ed. Majumder, Mitra and Parui, ACM Digital Library, Dec 2006.
15. जिज्ञासु, ब्रह्मदत्त, 1979. अष्टाध्यायी-भाष्य, 3 भाग, रामलाल कपूर ट्रस्ट, बहालगढ सोनीपत, हरियाणा
16. Joshi, S.D. (editor) 1968. Patanjali's Vyakarana Mahabhashya}, (several volumes), Univ. of Poona, Pune, India.
17. Joshi, S.D. and Roodebergen J.A.F., 1998. The A▪▪ādhyāyī of Pā▪ini (several volumes), Sahitya Akademi, Delhi, India.
18. Kiparsky, P. , On the Architecture of Panini's Grammar, Proceedings of the Sanskrit Computational Linguistics Symposium, Ed. Huet, Kulkarni and Scharf, LNAI 5402, Springer Verlag, 2009.
19. Lin D. , Dependency-based evaluation of MINIPAR. In Workshop on the evaluation of Parsing Systems, Granada, Spain, 1998.
20. Marneffe M., MacCartney B. and Manning C. D. , Generating Typed Dependency Parses from Phrase Structure Parses, The fifth international conference on Language Resources and Evaluation, L, REC 2006, Italy.
21. Ramakrishnamacharyulu, K.V., Annotating Sanskrit Texts based on śābdabodha systems, Proceedings of the Sanskrit Computational Linguistics Symposium, Ed. Kulkarni and Huet, LNAI 5406, Springer Verlag, 2009.
22. Staal, J.F., Word Order in Sanskrit and Universal Grammar, Reidal, Dordercht (Foundations of Language, Supplementary series: v.5), 1967.
23. Sleator D. D., Temperley D., Parsing English with a link grammar, In third international Workshop on Parsing Technologies, 1993.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

वाक्यविश्लेषके प्रयुक्तवाक्यानि

(1) नद्याः तटे एकः वृक्षः अस्ति। (2) वृक्षस्य समीपम् एका शिला अस्ति। (3) वृक्षस्य शाखासु नीडाः सन्ति। (4) नीडेषु विहगाः वसन्ति। (5) नीडाः विहगान् रक्षन्ति। (6) वृक्षस्य अधः वानराः सन्ति। (7) कपयः गृहम् न रचयन्ति। (8) ते सर्वदा इतस्ततः भ्रमन्ति। (9) एकस्मिन् दिवसे शीतम् तान् पीडयति। (10) ते शीतात् त्राणाय अग्निम् इच्छन्ति। (11) कुत्रापि ते अग्निम् न विन्दन्ति। (12) एकः गुञ्जायाः फलानि पश्यति। (13) गुञ्जायाः फलानि रक्तानि सन्ति। (14) सः अग्नेः सदृशानि गुञ्जायाः फलानि आनयति। (15) तानि गुञ्जायाः फलानि शिलायाम् संहरति। (16) ते सर्वे गुञ्जा-फलम् परितः उपविशन्ति। (17) अग्नेः इच्छया ते मुखैः तानि धमन्ति। (18) ते अग्निम् न विन्दन्ति। (19) ते वानराः अनलाय वृथा आयासम् कुर्वन्ति। (20) तेषाम् शीतम् न नश्यति। (21) कपयः मूर्खाः सन्ति।

वाक्यविश्लेषके प्रयुक्तसम्बन्धाः

0	उपपदविभक्तिः	12	कालाधिकरणम्
1	कर्ता	13	विषयाधिकरणम्
2	प्रयोजककर्ता	14	कर्तासम्बन्धिकरणम्
3	प्रयोज्यकर्ता	15	विशेषणम्
4	कर्म	16	क्रियाविशेषणम्
5	वैपरीत्येन गौणकर्म	17	तादर्थ्यम्
6	वैपरीत्येन मुख्यकर्म	18	पूर्वकालीनम्
7	करणम्	19	सम्बन्धः
8	सम्प्रदानम्	20	कारक-षष्ठी
9	अपादानम्	21	निषेध
10	अधिकरणम्	22	सम्बोधन
11	देशाधिकरणम्		

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri

Evaluating Tagsets for Sanskrit

Madhav Gopal[1], Diwakar Mishra[2] and Priyanka Singh[3]

[1]Center for Linguistics, SLL & CS,
[2]Special Center for Sanskrit Studies,
[3]Centre for Indian Languages, SLL & CS,
Jawaharlal Nehru University, New Delhi

{mgopalt, diwakarmishra, devisi.translation} @gmail.com

Abstract. In this paper we present an evaluation of available Part Of Speech (POS) tagsets designed for tagging Sanskrit and Indian languages which are developed in India. The tagsets evaluated are - JNU-Sanskrit tagset (JPOS), Sanskrit consortium tagset (CPOS), MSRI-Sanskrit tagset (IL-POST), IIIT Hyderabad tagset (ILMT POS) and CIIL Mysore tagset for the Linguistic Data Consortium for Indian Languages (LDCIL) project (LDCPOS). The main goal behind this enterprise is to check the suitability of existing tagsets for Sanskrit from various Natural Language Processing (NLP) points of view.

Keywords: Aṣṭādhyāyī, POS tagging, POS tagger, tagset, morphology, Pāṇini, WSD, machine learning.

1 Introduction

POS tagging assigns a suitable part of speech label for each word in a sentence of a language. In other words, POS tagging is the process of identifying lexical/grammatical category of a word according to their function in the sentence. For NLP tasks, annotated corpora of a language have a great importance. Annotated corpora serve as an important resource for such well-known NLP tasks as POS-Tagging, chunking, parsing, structural transfer, word sense disambiguation (WSD) etc. POS tagging is also referred to as morpho-syntactic tagging or annotation. A POS Tagger (POST) is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc., although generally computational applications use more fine-grained POS tags like 'noun-masculine' etc.

The quality of the POS annotation in a corpus is crucial for the development of POST. The design of the annotated Sanskrit corpus is rapidly going on in India. Natural languages are intrinsically very complicated and Sanskrit is not an exception to this. Sanskrit is morphologically and lexically very rich language. It has a variety of words, lexemes, morphemes, and a rich productive mechanism of forming new

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

70

words. Due to identical inflections for various cases, many of its words are ambiguous and their disambiguation for NLP tasks is a must. Moreover, there are many factors involved in causing ambiguity in this language. A tagged Sanskrit corpus could be used for a wide variety of research like developing POS Taggers, chunkers, parser, WSD etc.

A tagged text corpus is useful in many ways. More abstract levels of analysis benefit from reliable low-level information, e.g. parts of speech, so a good tagger can serve as a preprocessor. It is also useful for linguistic research for example to help find instances or frequencies of particular constructions in large corpora. Apart from this it is good for stemming in information retrieval (IR), since knowing a word's part of speech can help tell us which morphological affixes it can host. Automatic POS taggers can help in building automatic word-sense disambiguating algorithms; POS taggers are also used in advanced ASR language models such as class-based N-grams. POS tagging is also useful for text to speech synthesis and alignment of parallel corpora.

2 Characteristics of the Sanskrit language

Sanskrit is one of the most studied languages of the world. It has been analyzed through many schools of grammar in India and abroad as well. Traditionally, the number of grammatical categories of Sanskrit varies from one to five. According to Indra school of *vyakaraṇa* there is only one category of words (*arthaḥ padam*). Paṇini recognizes two categories: nominal and verbal (*suptiṅantam padam* - Pāṇini 1.4.14). The Nyaya philosopher Jagadishạ in his *śabdaśaktiprakaśika* accepts only three categories of words: base, suffix and particles (*prakṛti*, *pratyaya* and *nipata*). The most popular categorization is of Yaska who describes the language in four categories: nominal base, verb root, prefix and indeclinable (*catvāri padajātāni nāmākhyate copasarganipātāśca* – Nirukta 1.1). The new grammarians (navya vaiyākaraṇa) add one more category to the list of Yaska, that is, *karmapravacaniya* and accept five categories of words. The different views on number of word categories are given in Durga's commentary on *Nirukta* [19]. If we take Yaska's grammatical categories for POS tagging, it will be very coarse grained tagging. But it will help us give high accuracy result as less number of tags lead to efficient machine learning. Moreover, two of these categories (prefix and indeclinable) belong to the closed class and the remaining two categories have totally different set of inflectional suffixes which are relatively easily identifiable. Apart from this, the set of roots of one category (verb) is also a closed list, that is, *dhātupāṭha* (list of verb roots listed by Pāṇini).With so shallow information this accuracy will not be of much help in many kinds of NLP tasks like in WSD which requires very fine-grained analysis.

Sanskrit is a relatively free word order language. In Sanskrit, a syntactic unit is called *pada*. Cardona [3] posits the formula for Sanskrit sentence (N-En)p…(V-Ev)p. A *pada* can be a nominal (*subanta*) or verbal (*tiṅanta*) expression. *Padas* with *sup* (nominal) inflections constitute the NPs (*subanta padas*), and those with *tiṅ* (verbal)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

71

can be called constituting the VPs (*tiṅanta pada* along with object *subantas*). In the former, the bases are called *prātipadikas* which undergo *sup* suffixations under specifically formulated conditions of case, gender, number, and also the end-characters of the bases to yield nominalsyntactic words. The rules for *subanta padas* are found scattered in *Aṣṭādhyāyī* mostly in chapters 7-1, 7-2, 7-3, 6-1, 6-4. However, these rules have been treated in the subanta chapter of *Siddhānta Kaumudī* from rule number 177 to 446 [10].

The derivational morphology in Sanskrit studies primary forms (*kṛdanta*) and secondary forms (*taddhita*), compounds (*samāsa*), feminine forms (strīpratyayānta) etc. These can be inflected for 21 case (7 cases x 3 number) affixes to generate 21 inflected forms.

The verb morphology (*tiṅanta*) is equally complex. Sanskrit has approximately 2014 verb roots including *kaṇvādi* according to Pāṇinian *dhātupāha* classified in 10 gaṇas to undergo peculiar operations; it can also be subclassified in 12 derivational suffixes. A verb root conjugates for tense, mood, number and person information. Further, these can have *ātmanepadī* and *parasmaipadī* forms in 10 *lakāras* and 3x3 person and number combinations. There are12 secondary suffixes added to verb roots to create new verb roots. A verb root may have approximately 2190 (tense, aspect, number etc.) morphological forms. Mishra et al. [17] have done a rough calculation of all potential verb forms in Sanskrit to be more than 1029,60,000.

3 Evaluation Discussion of Tagsets

This section presents a survey of tagsets designed specifically for Sanskrit or for all Indian languages and then their evaluation with respect to Sanskrit. The tagsets are given in tables in their respective sub-sections.

3.1 ILMT Tagset (IIIT-Hyderabad)

NLP researchers at IIIT-Hyderabad have designed a standard POS tagset called ILMT tagset, guidelines for POS tagging and Chunking for Indian Languages and also brought out a collection of Application Program Interfaces (APIs) and tools for NLP called *Sanchay*, which also includes syntactic annotation interface. Their tagset is a flat one derived from the Penn tagset for English. They have made certain changes in the original tagset to make it suitable for Indian languages. Wherever the Penn tags were found to be inadequate for Indian languages, either new tags were introduced or existing tags were modified. It has less number of tags with the assumption that having less tags will facilitate machine learning and will give higher accuracy and coverage. This tagset covers mainly lexical categories and does not consider syntactic function of the word in the sentence, citing a reason that this reduces confusion involved in manual tagging. The machine is able to establish a word-tag relation which leads to efficient machine learning. The following table contains all the tags listed in this tagset:

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

(72)

NN	Noun	CC	Conjunct
NNP	Proper Noun	QW	Question Words
PRP	Pronoun	QF	Quantifier
VFM	Verb Finite Main	QFNUM	Number Quantifiers
VAUX	Verb Auxiliary	INTF	Intensifier
VJJ	Verb NonFinite Adjectival	NEG	Negative
VRB	Verb NonFinite Adverbial	NNC	Compound Common Nouns
UNN	Verb NonFinite Nominal	NNPC	Compound Proper Nouns
JJ	Adjective	NVB	Noun in kriya mula
RB	Adverb	JVB	Adjective in kriya mula
NLOC	Noun Location	RBVB	Adv in kriya mula
PREP	Postposition	UH	Intrjection Words
RP	Particle	SYM	Special

Table 1: ILMT Tagset

Thus, this tagset allows only coarse linguistic analysis and avoids fine-grained distinctions among linguistic items. At the same time, the designers of the tagset believe that "too coarse an analysis is not of much use". They try to seek a balance between fineness and coarseness. According to them, analysis should not be so fine as to hamper machine learning and also should not be so coarse as to lose important information. This tagset does not include attributes of lexical categories such as gender, number, person etc. Apart from this the tags for incorporated adjective and adverb are not applicable in Sanskrit. In Sanskrit, like noun and verb, Particles also require more tags as they have different and valuable function in the language, so in order to suit Sanskrit this tagset will have to be modified significantly. Keeping in mind the complex morphology of Sanskrit this flat tagset will hardly be able to do WSD.

3.2 JNU-Sanskrit Tagset (JPOS)

R. Chandrashekar's doctoral thesis titled *Part-of Speech Tagging for Sanskrit* [4] deals with the POS tagging problem for Sanskrit for the first time. In the linguistic analysis, JPOS follows the Pāṇinian grammatical description very strictly. His classification of Sanskrit linguistic items is very exhaustive and it provides a very unique tag applying the Sanskrit grammatical terms as much as possible. In the following table the tag description of main categories is given:

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

Common Noun (*nāmapada*)	N
Proper noun	NA
Proper Noun Country (*nāma abhidhāna deśa*)	NAD
Patronymic Noun	NAP
Metronymic Noun	NAS
Proper Noun Nationality (*nāma abhidhāna tadrāja*)	NAT
Noun Desiderative	NS
Coordinative Enumerative Compound (*dvandva itaretara*)	NCDI
Coordinative Collective Compound (*dvandva saṃhara*)	NCDS
Determinative Compound Accusative	NCT2
Determinative Compound Instrumental	NCT3
Determinative Compound Dative	NCT4
Determinative Compound Ablative	NCT5
Determinative Compound Genitive	NCT6
Determinative Compound Locative	NCT7
aluk Compound	NCAl
Negative Compound	NCNT
karmadhāraya Compound	NCK
dvigu Compound	NCD
bahuvrīhi Compound	NCB

Adverbial Compound	NCA
Pronoun	SN
Pronoun First Person	SNU
Pronoun Second Person	SNM
Pronoun Reflexive	SNA
Pronoun Demonstrative	SNN
Pronoun Interrogative	SNP
Pronoun Relative	SNS
Parasmai Pada Verb	AV
Atmane Pada Verb	PV
Cardinal Number	SAM
Ordinal Number	SAMY
Past Active Participle	KV1
Past Middle/Active Participle	KV2
Past Passive Participle	KB1
Past Active Participle	KB2
Future Active Participle	KAa
Future Passive Participle	KAb
kṛdanta vidhyarthaka 1	KVI1
kṛdanta vidhyarthaka 2	KVI2
kṛdanta vidhyarthaka 3	KVI3
Particles	AV
Negative Particle	AVN
Conjunctive Particle	AVC
Disjunctive Particle	AVD
Interrogative Particle	AVP
Infinitive Particle	AVT
avyaya ktvānta	AVK
avyaya lyabanta	AVL
Adverb Particle	AVKV
Interjection Particle	UD

Table 2: JNU-Sanskrit Tagset

In this tagset, there are 65 word class tags, 43 feature sub-tags, and 25 punctuation tags and one tag UN to annotate unknown words – a total of 134 tags. A single full tag is a combination of word class tag and feature sub-tags (indeclinable and punctuation tags do not have sub-tags). This is helpful in ambiguity resolution as person and number sub-tags have an agreement between the nouns and verbs in Sanskrit. The complete table for verb tags is not given here; I have given only two

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

(74)

tags for it- *Parasmai Pada* Verb and *Atmane Pada* Verb. Verb takes *lakāra*, and person-number tags. Adjectives are included with their degrees, that is, they have 3 tags.

This is a flat and fine-grained tagset capturing linguistic features according to the traditional Sanskrit grammar. Its coverage is very high. Almost every kind of linguistic item was treated discretely (especially *taddhita* and *kṛidanta* constructions). This tagset is very informative but we assume that its specificity will suffer while tagging the data. For instance, among noun tags there are 8 tags and the basis of counting them separately seems to be the special morphology for these items like Patronymic Noun (e.g. *rāghavaḥ, jānakī*), Metronymic Noun (e.g. *pārthaḥ, saumitriḥ*), (both tags could be treated as Proper Noun), Agent Noun (e.g. *kartuḥ, adhyāpakaḥ*) (could be treated as Common Noun). Likewise under Compound category there are 14 types. Many of which could be clubbed under other categories as well. For example, the words like *rājapurusaḥ* and *devendraḥ* have been assigned Determinative Compound (genitive) which could be very well given Common noun and Proper noun respectively. The division of participle tags is also very lengthy; *kṛidanta vartamāna* has two separate tags which does not seem serving any purpose and the same is with *kṛidanta vidhyurthaka* which has 3 tags.

Apart from this, the human annotator will have enormous cognitive load as this tagset requires remembering this huge number of tags along with exhaustive knowledge of Pāṇinian grammar in order to apply them correctly. Only a sophisticated pundit of *Pāṇinian* grammar can handle this task. At experiment level, however, no data have been tagged to test its efficiency, so the above is our assumption only. Moreover, its adaptation for other Indian languages will invite enormous changes in the tagset. So the tagged multilingual corpora will hardly be compatible with each other.

3.3 LDCIL Tagset

Linguistic Data Consortium for Indian Languages (LDC-IL) is a consortium for the development of language technology in India and was set up by a collective effort of the Central Institute of Indian Languages, Mysore and several other like-minded institutions working on Indian Languages technology. LDCIL tagset is available in the website of Linguistic Data Consortium for Indian Languages: http://www.ldcil.org/POSTagSet.html. This tagset was formulated by the Standardization Committee on August 6, 2007 at IIIT, Hyderabad and is a further extension of ILMT tagset with some additional tags to achieve higher level of granularity. It is a standard tagset for annotating the corpora of Indian languages. It is a flat tagset designed in the pattern of ILMT tagset and has a total of 26 tags. The table below contains all the tags.

Noun	NN	Proper Noun	NNP
NLOC	NST	Compound (any)	XC

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

75

Pronoun	PRP
Demonstrative	DEM
Verb Finite	VM
Verb Auxiliary	VX
Adjective	ADJ
Adverb	ADV
Postposition	PSP
Particles	PRT
Conjuncts	CNJ
Question Words	WQ
Quantifiers	QTF
Cardinal	CRD
Ordinal	ORD
Classifier	CL
Intensifier	INTF
Interjection	INJ
Negation	NEG
Quotative	UT
Symbol	SYM
Reduplicative	RDP
Echo word	ECH
Unknown	UNK

Table 3: LDCIL Tagset

This tagset is similar to ILMT tagset, so the same evaluation goes to this also.

3.4 IL-POSTS Sanskrit Tagset

This tagset has been derived from IL-POSTS, a standard framework for Indian languages developed by MSRI. This is a hierarchical tagset which is based on guidelines similar to EAGLES. This tagset encodes information at three levels- Categories, Types, and Attributes. Till date (as far as we know) four language specific tagsets have been derived from this framework; Hindi, Bangla, Tamil and Sanskrit. The tagset for Sanskrit is given here in these two tables- one having categories and their respective types and the other containing attributes and their respective values:

Category	Type
Noun (N)	Common (NC)
	Proper (NP)
Verb (V)	
Pronoun (P)	Pronominal (PPR)
	Reflexive (PRF)
	Reciprocal (PRC)
	Relative (PRL)
	Wh (PWH)
Nominal Modifier (J)	Adjective (JJ)
	Quantifier (JQ)
Demonstrative	Absolutive (DAB)
	Relative (DRL)
(D)	Wh- (DWH)
Kridant (KD)	Participle (KDP)
	Gerundive (KDG)
Particle (C)	Coordinating (CCD)
	Subordinating (CSB)
	Gerundive (CGD)
	Interjection (CIN)
	Negative (CNG)
	Emphatic (CEM)
	Interrogative (CNT)
	Adverb (CAD)
	Ambiposition (upapada) (CPP)
	Quotative (CQT)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

76

	Comparative (CCM)
	Reduplicative (CRD)
	Other (CX)
Punctuation (PU)	
Residual (RD)	Foreign word (RDF)
	Symbol (RDS)
	Others (RDX)

Table 4.1: Categories and their types

Attributes	Values
Gender (Gen)	Masculine, Feminine, Neuter
Number (Num)	Singular, Dual, Plural
Person (Per)	First, Second, Third
Case (Cs)	Nominative, Accusative, Instrumental, Dative, Ablative, Genetive, Locative, vocative
Nominaldeclension vibhakti (Vbh)	prathamā, dvitīyā, tṛitīyā, caturthī, pañcamī, ṣaṣṭhī, saptamī, vocative
Tense/Mood (Tns/Mood)	Present, Aorist, Imperfect, Perfect, Periphrastic Future, General Future, Imperative, Potential, Benedictive, Conditional
Numeral (Nml)	Ordinal, Cardinal, Non-numeral
Distance (Dist)	Proximal, Distal
Emphatic (Emph)	Yes y, No n
Negative (Neg)	Yes y, No n
Honorificity (Hon)	Yes y, No n
Pada (Pd)	parasmaipada, ātmanepada

Table 4.2: Attributes and their values

The standard which has been followed in this tagset takes care of the linguistic richness of Indian languages including Sanskrit. It allows the "selective inclusion and removal of features for a specific language/project, thereby keeping the framework a common standard across languages/projects" [1]. This tagset is supposed to provide cross-linguistic compatibility, reusability, and interchangeability. Keeping in mind its higher accuracy results, it could prove a boon to Indian NLP which still needs to achieve a good position in the world as Indian languages are lacking adequate resources in terms of data and tools (see [1] and [11]).

This framework is very sophisticated in covering the linguistic features of Indian languages. The tags used are extremely fine-grained, and incorporate a great deal of information about case, gender, number and so on. The IL-POST is able to accommodate all desired linguistic features of Sanskrit and the tagged corpus remains compatible with other languages tagged by a brethren tagset. Apart from this, as this tagset follows the standard of European languages it could enable us to access the European language data for Indian NLP and vice versa.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

77

MSRI has developed an annotation tool called *MSRI Part-of-Speech Annotation Interface* which is a GUI for assigning the appropriate POS label for each word in a sentence. This is designed for annotating within the framework of a hierarchical tagset designed at MSRI. The tool also provides the facility to provide the morphological attributes with their values. The interface supports various operations for viewing and editing the POS labels and morphological information associated with the word. This tool reduces the cognitive load of the human annotator and enables one to speed up the tagging task.

3.5 Sanskrit Consortium Tagset (CPOS)

This tagset is designed for tagging sandhi-free Sanskrit corpus. It is based on traditional Sanskrit grammatical categorization. All the tags seem to be the extensions of Yāska's four grammatical categories: nāma, ākhyāta, upasarga and nipāta. This tagset has 28 tags. All the tags are given in the following table:

NP	nāmapada (common noun)
NS	nāmasanjnā (proper noun)
NV	nāmaviśesan (adjective)
KN	kridantanāma (participles)
KNV	kridantanāma viśeṣaṇa (participle adjective)
SN	sarvanām (pronoun)
SNV	sarvanām-viśeṣaṇa (demonstrative)
SSN	sāmbandhika sarvanām (relative pronoun)
SMK	saṃkhyavācaka (number)
SMKY	saṃkhyeyavācaka (cardinal)
SMKP	saṃkhyāpūraṇavācaka (ordinal)
KP	kriyāpada (verb)
A	avyaya (indeclinable)

AN	niṣedhārthaka (negation)
AP	praśnārthaka (interrogative)
AUP	upapada (ambiposition)
ASMB	saṃbodhanaṃ (vocative)
AUD	uddharaṇa (quotative)
KKS	Gerundive
KV	kriyāvišesan (adverb)
ABS	(udgāra) interjection
APY	pada yojaka (connective)
AVK	vakya yojaka (sentential connective)
DV	dwirukti (reduplication)
ASD	sādṛśyādi (comparative)
AB	other language

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

78

UNK	Unknown	PUNC	Punctuation

Table 5: Sanskrit Consortium Tagset

This is apparently a flat tagset and allows the annotation of major categories only. Most of the categories appear to have been adapted either from the JPOS or the ILMT tagset. For morphological analysis it will take help from Morphological Analyzer, so morpho-syntactic features are not included in the tagset. It is supposed to give higher accuracy result by having less number of tags. A tool called *Sanchay* is being used to facilitate manual tagging and thus it reduces the cognitive load of annotator. For Sanskrit tagging, it can be useful as long as we know what will be the use of tagged data in this tagset. At an experimental level two taggers were developed using this tagset - a rule based and a statistical and have reported 79-81% results. The rule based tagger can be accessed using this link http://sanskrit.jnu.ac.in/cpost/post.jsp. As this tagset does not follow any standard framework the tagged data will hardly allow any sharing and reusability. Due to its coarse-grained tagging and unavailability of Indian language morph analyzers, it will be interesting to see how this tagset can be put to actual use.

4 Conclusion and Further Research

From the above evaluation of the tagsets it appears that the most useful tagged corpus is the one that ensures the maximal use and sharing of it, and it will only be possible when it is tagged by a well-established standard tagset or paradigm. The tagsets that we have discussed in this article are of both kinds- those following global standards/guidelines and those which have evolved from particular language specific requirements. The tagsets like ILPOST, ILMT and LDCIL are designed to take care of all Indian languages, and this could provide resource sharing and reusability of the linguistic resources. The flat tagsets may be easier to process but they cannot capture higher level of granularity without an extremely large list of independent tags. They are hard to adapt too. So far, ILPOSTS has been used for four languages Hindi, Tamil, Bangla and Sanskrit, and an adapted version is reportedly being used in the LDC-IL project at CIIL Mysore for many more languages. So, in order to enrich Indian NLP resources, ILPOSTS seems to be a better option. However, further research is needed to substantiate this claim.

For further research a certain amount of data could be manually tagged with the help of these tagsets by some (1 or 2) human annotator. And on the basis of the experience of these annotators certain facts can be discovered, for example, which tagset is easy to use for Sanskrit and which tagset gives better results and which is less confusing and so on. Thereafter, rule based or statistical taggers can be tested for the efficiency of these tagsets and also their compatibility across languages.

—

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

79

References

1. Baskaran, S., Bali, K., Bhattacharya, T., Bhattacharyya, P., Choudhury, M., Jha, G.N., Rajendran, S., Saravanan, K., Sobha L., Subbarao, K.V.S.: A Common Parts-of-Speech Tagset Framework for Indian Languages. LREC, Marrakech, Morocco (2008)
2. Baskaran, S. et al.: Framework for a Common Parts-of-Speech Tagset for IndicLanguages (2007), http://research.microsoft.com/~baskaran/POSTagset
3. Cardona, G.: Pāṇini: His work and its traditions. Motilal Banarasidass, Delhi (1988)
4. Chandrashekar, R.: Parts-of-Speech Tagging For Sanskrit. Ph.D. thesis submitted to JNU, New Delhi (2007)
5. Greene, B.B. and Rubin, G.M.: Automatic grammatical tagging of English. Providence, R.I.: Department of Linguistics, Brown University (1981)
6. Hardie, A.: The Computational Analysis of Morphosyntactic Categories in Urdu. PhD Thesis submitted to Lancaster University (2004)
7. Hellwig, O.: SANSKRITTAGGER, A Stochastic Lexical and POS Tagger for Sanskrit. In: Huet, G. & Kulkarni A. (eds.) Proceedings of the First International Symposium on Sanskrit Computational Linguistics (2007)
8. Huet Gerard, The Sanskrit Heritage Site, http://sanskrit.inria.fr/
9. IIIT-Tagset. A Parts-of-Speech tagset for Indian Languages. http://shiva.iiit.ac.in/SPSAL2007/iiit_tagset_guidelines.pdf
10. Jha, Girish N.: Generating nominal inflectional morphology in Sanskrit. *SIMPLE 04*, IIT-Kharagpur Lecture Compendium, Shyama Printing Works, Kharagpur, (2004)
11. Jha, Girish N., Gopal, M., Mishra, D.: Annotating Sanskrit Corpus: adapting IL-POSTS. In: Z. Vetulani (ed.) Proceedings of the 4th Language and Technology Conference: Human Language Technologies as a challenge for Computer Science and Linguistics, pp. 467-471 (2009)
12. Jha, Girish N., Mishra, S.: Semantic processing in Panini's karaka system, *Lecture Notes in Computer Science series*, 5402 Springer, Berlin/ Heidelberg (2009)
13. Kale, M.R.: A Higher Sanskrit Grammar. MLBD Publishers, New Delhi (1995)
14. Leech, G and Wilson, A.: Recommendations for the Morphosyntactic Annotation of Corpora. EAGLES Report EAG-TCWG-MAC/R (1996)
15. Leech, G and Wilson, A.: Standards for Tag-sets. In: Hans van Halteren (ed.) Syntactic Word class Tagging. Kluwer Academic, Dordrecht (1999)
16. Leech, G.: Grammatical Tagging. In: Garsire, Leech, and McEnery (eds.) Corpus Annotation: Linguistic Information for Computer Text Corpora. Longman, London (1997)
17. Mishra S., and Jha, Girish N.: Identifying verb inflections in Sanskrit morphology. In: proceedings of SIMPLE'04, IIT Kharagpur, (2005)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

(80)

18. Ramkrishnamacharyulu, K.V.: Annotating Sanskrit Texts Based on Sabdabodha Systems. In: Kulkarni, A. and Huet G. (cds) Proceedings of the Third International Symposium on Sanskrit Computational Linguistics, pp. 26-39 (2009)
19. Rishi, U. S. S. (ed.), '*Yaska-pranitam niruktam*', Vol. I. Chowkhamba Vidyabhawan, Varanasi, (2005)
20. Santorini, B.: Part-of-speech tagging guidelines for the Penn Treebank Project. Technical report MS-CIS-90-47, Dept. Of Computer and Information Science, University of Pennsylvania, (1990)

Appendices:

Appendix 1 Sample of Tagged data by Sanskrit Consortium tagset

दमनकः\NS आह\KP -\PUNC "\PUNC यद्यपि\AVK न\AN किंचित्\SN देवपादानाम्\NP अस्माभिः\SN प्रयोजनम्\NP ,\PUNC तथापि\AVK भवताम्\SN प्राप्तकालम्\NV वक्तव्यम्\KN ,\PUNC यत्\SSN उत्तममध्यमाधमैः\NV सर्वैः\SN अपि\ASD राज्ञाम्\NP प्रयोजनम्\NP ।\PUNC उक्तम्\KN च\AVK -\PUNC

दन्तस्य\NP निष्कोषणकेन\NP नित्यम्\KV कर्णस्य\NP कण्डूयनकेन\NP वा\AVK अपि\A ।\PUNC
तृणेन\NP कार्यम्\NP भवति\KP ईश्वराणाम्\NP किमङ्ग\ABS वाग्घस्तवता\NV नरेण\NP ॥\PUNC 77\PUNC ॥\PUNC

तथा\AVK वयम्\SN देवपादानाम्\NP अन्वयागताः\NV भृत्याः\NP आपत्सु\NP अपि\ASD पृष्ठगामिनः\NV यद्यपि\AVK स्वम्\NV अधिकारम्\NP न\AN लभामहे\KP ,\PUNC तथापि\AVK देवपादानाम्\NP एतत्\SN युक्तम्\NV न\AN भवति\KP ।\PUNC

Appendix 2 Sample of Tagged data by IL-POST tagset

दमनकः\NP.mas.sg.nom.i आह\V.ppd.sg.3.prs.n -\PU "\PU यद्यपि\CSB न\CNG किंचित्\CX देवपादानाम्\NC.mas.pl.gen.vi अस्माभिः\PPR.0.pl.1.ins.iii.n.n.prx प्रयोजनम्\NC.neu.sg.0.i ,\PU तथापि\CSB भवताम्\PPR.mas.pl.3.gen.vi.n.n.prx प्राप्तकालम्\JJ.neu.sg.0.i.n.n वक्तव्यम्\KDG.neu.sg.0.i ,\PU यत्\CSB उत्तममध्यमाधमैः\JJ.neu.pl.ins.iii.n.n सर्वैः\PPR.neu.pl.3.ins.iii.n.n.0 अपि\CEM राज्ञाम्\NC.mas.pl.gen.vi प्रयोजनम्\NC.neu.sg.0.i ।\PU उक्तम्\KDP.neu.sg.0.i च\CCD -\PU

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

(81)

दन्तस्य\NC.mas.sg.gen.vi निष्कोषणकेन\NC.mas.sg.ins.iii नित्यम्\CAD कर्णस्य\NC.mas.sg.gen.vi कण्डूयनकेन\NC.mas.sg.ins.iii वा\CCD अपि\CEM ।\PU

तृणेन\NC.neu.sg.ins.iii कार्यम्\KDG.neu.sg.0.i भवति\V.ppd.sg.3.prs.n ईश्वराणाम्\NC.mas.pl.gen.vi किमङ्ग\CIN वाग्धस्तवता\JJ.mas.sg.ins.iii.n.n नरेण\NC.mas.sg.ins.iii ॥\PU 77\RDS ॥\PU

तथा\CCM वयम्\PPR.0.pl.1.nom.i.n.n.prx देवपादानाम्\NC.mas.pl.gen.vi अन्वयागताः\JJ.mas.pl.0.i.n.n भृत्याः\NC.mas.pl.0.i आपत्सु\NC.mas.pl.loc.vii अपि\CEM पृष्ठगामिनः\JJ.mas.pl.0.i.n.n यद्यपि\CSB स्वम्\PRF.mas.sg.acc.ii अधिकारम्\NC.mas.sg.acc.ii न\CNG लभामहे\V.apd.pl.1.prs.n ,\PU तथापि\CSB देवपादानाम्\NC.mas.pl.gen.vi एतत्\PPR.neu.sg.3.nom.i.n.n.prx युक्तम्\JJ.neu.sg.0.i.n.n न\CNG भवति\V.ppd.sg.3.prs.n ।\PU

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

Indian Linguistics Dec 2009

Sanskrit Morphological Analyser: Some Issues

Amba Kulkarni and Devanand Shukl

Department of Sanskrit Studies,
University of Hyderabad,
Hyderabad
apksh@uohyd.ernet.in,dev.shukl@gmail.com

1 Introduction

Sanskrit has rich inflectional as well as derivational morphology. At the same time it enjoys the privilege of having a formal (to a large extent) grammar in the form of Aṣṭādhyāyī built over 25 centuries ago. One may think, therefore, it would be a trivial task to build a morphological analyser based on this grammar. But it is not so. Though well defined rules for Sanskrit morphology exist in Aṣṭādhyāyī, for a typical computational linguist without any knowledge of Sanskrit, it is difficult to build a system incorporating these rules. At the same time, it is rare to find a person who understands the Aṣṭādhyāyī well and has also a good knowledge of computers for its implementation. Further there are controversies over the use of meta rules (paribhāṣā) used for application of the rules of Aṣṭādhyāyī. Therefore, incorporating the rules of Aṣṭādhyāyī as they are, is a difficult task. As a result, in spite of the existence of a formally defined and well described grammar, construction of a set of computational tools for the analysis of Sanskrit texts could not take a momentum for a long time.

If we look at the Sanskrit literature, we see that many attempts have been made to render the learning of Sanskrit word formation easier. The Sanskrit grammatical tradition provides evidence for the implementation of the two well known models of morphological descriptions viz. Item and Process model and the Word and Paradigm model. While the rules in Aṣṭādhyāyī may be best described as an example of Item and Process model, the later Sanskrit texts such as Siddhānta-Kaumudī[15], Rūpacandrikā (a compilation of representative word forms), Dhāturatnākara[19] (a lexicon of verb forms), Kṛdanta-Rūpamālā[17] (a book listing frequently used nominal stems derived from the verbal roots) use the Word and Paradigm model and are best suited either for pedagogical purpose or for quick reference. Both these traditions are equally popular among the Sanskrit scholars and have their own advantages.

The issues related to the development of Sanskrit morphological analyser ranging from the representation of phonemes for computation, the suitability of Unicode versus various transliteration schemes for internal representation, handling sandhi and the complexity involved in building the analyser are well described

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

by Huet in [7],[8]. In this paper, we would like to model the morphology described by Pāṇini from a computational perspective, and show how one can take advantage of the various readily available databases described above to develop a morphological analyser quickly. Such an analyser then can be used to evaluate the performance of various other morphological analysers. The issues involved in comparing the analysers are discussed, and finally we suggest evaluation criteria to evaluate the morphological analysers.

2 Word Formation in Sanskrit

Figure given below gives a computer scientist's perspective of the Sanskrit word formation as described by Pāṇini. We deviate from Pāṇini at certain places and interpret the whole process from the meaning point of view rather than just looking at the process involved therein[1]. The basic inflectional morphology is described by the sūtra *suptiṅantam padam* (1.4.14). Thus in the figure below, the formation of verbal forms (tiṅantas) and the nominal forms (subantas) from the verbal roots (dhātus) and the nominal bases (prātipadikas) respectively, represents the inflectional morphology in Sanskrit. The list of non-derived verbal roots is more or less a closed list[2]. The list of nominal bases on the other hand is open[3]. *Sup* and *tiṅ* are the list of inflectional suffixes a nominal base and a verbal root take respectively.

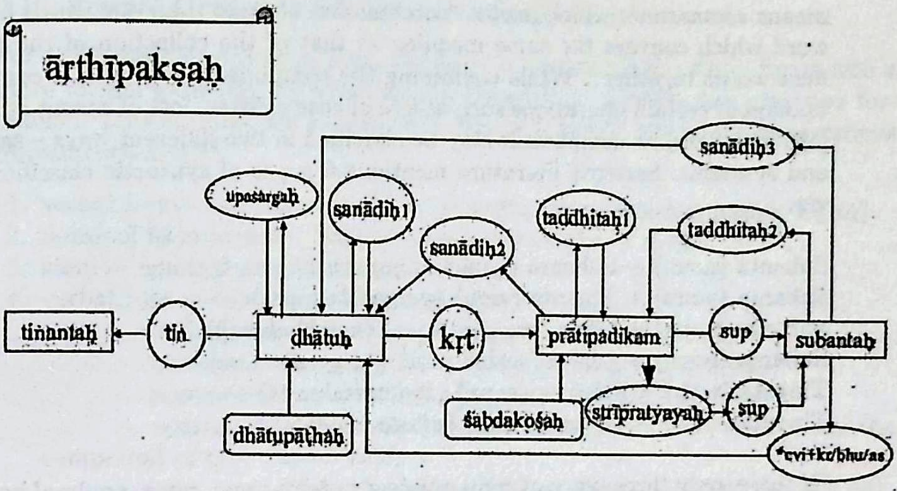
The *kṛt* and the *taddhita* are the derivational suffixes added to the verbal roots[4] and nominal bases[5] respectively, producing new nominal bases. *Sanādi* suffixes are of 3 types: some operate on nominal bases[6], some on nouns[7] and some operate on verbal roots[8]. Prefixes when attached to the verbal roots change the meaning of verbal roots and inflect further like a verb[9]. Finally there are certain nominal forms which take a special suffix *cvi* followed by one of the verbal roots *kṛ*, *bhū* or *as* to produce new verbal bases[10].

It is obvious from the given figure that the recursion in this word formation diagram leads to over-generation. But as is evident from the Pāṇinian sūtras, there

1 From processing point of view first the *tiṅ* is attached and then the upasarga gets attached to the verb form, whereas from the meaning point of view, verbal root together with the upasarga give a special meaning to the verb.
2 We find some instances of verbal roots which are not listed in the dhātupāṭha such as *ḍamb* from which the word *viḍambanā* is derived.
3 arthavat adhātuḥ apratyayaḥ prātipadikam (1.2.45)
4 (dhātoḥ) kṛt atiṅ (3.1.93)
5 kṛt taddhita samāsaśca (1.2.46)
6 lohitādiḍājbhaḥ kyaṣ 3.1.13
7 supaḥ ātmanaḥ kyac (3.1.8), etc.
8 sanādyanta dhātava.h (3.1.32)
9 prādayaḥ upasargaḥ kriyāyoge (1.4.58)
10 kṛbhvastiyoge saṃpadyakartari cviḥ (5.4.50)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.



Legends:	
tiṅantaḥ: verbal form	**tiṅ**: verbal suffix
prātipadikam: nominal base	**sup**: nominal suffix
kṛt: derivational suffix	**sanādiḥ**: derivational suffixes
upasargaḥ: verbal prefix	**śabdakoṣaḥ**: list of nominal bases
strīpratyayaḥ: feminine suffixes	**taddhitaḥ**: nominal derivational suffixes
subantaḥ: nominal form	**dhātupāṭaḥ**: list of verbal roots

is no mention of any limit over the recursion. Here are some examples showing recursion and productive nature of Sanskrit morphology.

1. pāṭhayitā = paṭh(verbal root) + ṇic(sanādiḥ1) + tṛc(kṛt) + feminine suffix + sup .
2. pipāṭhayiṣati = paṭh(verbal root) + ṇic(sanādiḥ1) + san(sanādiḥ1) + tiṅ .
3. vyāvahārī = vi(upasarga) + āṅ(upasarga) + ava(upasarga) + hṛ(verbal root) + ṇic(sanādiḥ1) + aṇ + feminine suffix + sup .
4. samabhivyāhāra = sam(upasarga) + abhi(upasarga) + vi(upasarga) + āṅ(upasarga) + hṛ(verbal root) + ghañ(kṛt) + sup .
5. susammodayantikā = su(upasarga) + sam(upasarga) + mud(verbal root) + ṇic(sanādiḥ1) + śatṛ(kṛt) + feminine suffix + kan(taddhita) + feminine suffix + sup .

However, the ability of human mind to process a complex string puts a limit on these potentially infinite productions to a finite number and is supported by the actual data. Words with more than three suffixes are typically rare compared to the words with single suffix or double suffixes. Further, for all practical purposes, the derived word forms can always be kept in the lexicon as headwords. These may be analyzed if there is a need.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

There is another important phenomenon which needs to be discussed viz. the compound formation in Sanskrit. The Sanskrit word *samāsaḥ* for a compound means *samasanam* which means "combination of more than one words into one word which conveys the same meaning as that of the collection of the component words together". While combining the components together, a compound undergoes certain operations such as loss of case suffixes, loss of accent etc. ([12], [3]). The Sanskrit compounds may be classified in two different ways – semantic and syntactic. Sanskrit literature mentions 6 types of syntactic classification[11] viz:

Subanta (noun) + Subanta (noun) (*rājapuruṣaḥ*)
Subanta (noun) + Tinanta (verb) (*paryyabhūṣayat*)
Subanta (noun) + nāma (nominal base) (*kumbhakāraḥ*)
Subanta (noun) + Dhātu (verbal root) (*kataprū*)
Tinanta (verb) + Subanta (noun) (*kṛntavicakṣaṇā*)
Tinanta (verb) + Tinanta (verb) (*pibata-khādata*)

Of these, only three types of compounds viz. noun-noun, noun-nonimal base and noun-verb(essentially the pre-verb verb sequence) are very frequent, and other 3 are rare. The rare compounds can always go in the lexicon. It is the productive compounds that need to be handled by the morphological analyser. Though compounds of 2 or 3 words are more frequent, compounds involving more than 3 constituent words with some compounds even running through pages is very common in Sanskrit literature. Here are some examples of Sanskrit compounds involving more than 3 words.

– veda-vedāṅga-tatva-jñaḥ
– pravara-mukuṭa-maṇi-marīci-mañjarī-caya-carcita-caraṇa-yugala
– jala-ādi-vyāpaka-pṛthivītva-abhāva-pratiyogi-pṛthivītva-vatī

Only the last component of the compound has the case suffix, and all the previous components assume a 'compounding form' – distinct from the nominal base. Thus a Sanskrit morphological analyser needs to analyse the compounding forms as well. Another phenomenon of Sanskrit compounds that needs to be handled by Sanskrit morpological analyser is the exo-centric(bahuvrīhi) compounds with the head or the final component exhibiting a gender of an object it denotes which may be different from its default gender. For example consider an exo-centric compound formed by the words *pīta* (yellow) and *ambaram* (cloth). The word *ambaram* is in neuter gender. When it is used to refer to a man wearing yellow clothes its form is *pītāmbaraḥ* and when it refers to a woman wearing yellow clothes its form is *pītāmbarā*. The head words *ambaraḥ* and *ambarā* are the nominative singular forms in masculine and feminine gender respectively. The default morphological analyser will fail to analyse these forms, *ambaram* being a neuter gender word.

[11] supāṁ supā tiṅā nāmnā dhātunātha tiṅāṁ tiṅā |
subanteti vijñeyaḥ samāsaḥ ṣaḍvidhoḥ budheḥ ||

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

2.1 Formal Grammar for Sanskrit Morphology

For the benefit of computational linguists, we translate the above figure into a more formal grammar by following the notation of context free grammar formalism. This exercise, we believe, will give us a better way to compare various implementations.[12]

1. verbal_form = verbal_root .[13] verbal_suffix (*(sup)tiṅantam padam* 1.4.14)
2. nominal form = stem . sup (*sup(tiṅ)antam padam* 1.4.14)
3. stem = nominal_stem + compound_stem
4. verbal_root = root (from Dhātupāṭha) +
   verbal_root . sanādiḥ1 (*sanādyanta dhātavaḥ* 3.1.32) +
   nominal_stem . sanādiḥ2 (*lohitādiḍājabhyaḥ kyaṣ* 3.1.13) +
   nominal_form . sanādiḥ3 (*sup ātmanaḥ kyac* 3.1.8) +
   upasargaḥ_seq . root (*upasargāḥ dhātuyoge* 1.4.58; Also covers the compound of type *supāṁ tiṅā*) +
   base . cvi . bhū,kṛ,as (*kṛbhvastiyoge sampadyakartari cviḥ 5.4.50)*
5. nominal_stem = nominal base + nominal_base . feminine_suffix
6. nominal_base = base (*arthavat adhātuḥ apratyayaḥ prātipadikam* 1.2.45) +
   base . taddhita (*(kṛt) taddhita (samāsaḥ) ca* 1.2.46) +
   verbal root . kṛt (*kṛt (taddhita samāsaḥ) ca* 1.2.46, and *dhātoḥ kṛt atiṅ* 3.1.96)
7. comp_stem = comp_base + compound_base . feminine_suffix
8. compound_base = pre_compound_base . nominal_base (*supāṁ supā*)
   pre_compound_base . {bhū,prū,..} (*supāṁ dhātu*) +
   pre_compound_base . {kāra/da/ja/...} (*supāṁ nāmnā*)
9. pre_compound_base = nominal_base . compound_suff +
   pre_compound_base . nominal_base . compound_suff

3 Various efforts

Attempts have been made in the past to develop morphological analysers for Sanskrit by the Center for the Development of Advanced Computing, India and Academy of Sanskrit Research, Melkote, India. However, they had either limited coverage or were almost unusable for any serious NLP applications. There have been notable efforts in the last few years in the area of Sanskrit computing. The efforts by Mishra[13], Scharf[16], Goyal[4], and Sridhar[18] are towards modelling Aṣṭādhyāyī using computers. They try to simulate the process of word generation following the rules of Aṣṭādhyāyī as closely as possible. One of the goals in their efforts is to understand 'how' much formal the system is and also to provide an environment so that various interpretations of Pāṇini's rules may be tested objectively. These implementations are still in evolving stage. Huet[7],[8]

12 We thank G. Huet for the valuable discussions on the Sanskrit Word Formation. These discussions not only helped us to improve the Sanskrit Word Formation figure substantially but also motivated us to express the grammar in a more formal way.

13 . is used to represent a joining operation and + indicates alternative possibilities

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

is more focussed on building a practical system rather than on the philosophical issues related to the structure of Aṣṭādhyāyī etc. Naturally he uses the modular finite state transducers to model the morphological phenomenon including the internal sandhi as well. The various inflected and derived forms are generated following the Word and Paradigm approach, and though is not in the strict sense Pāṇinian, captures the generalisation found in Pāṇini. Jha et al[10] have implemented the Sanskrit morphological analyser using database models wherein they store the suffixes and stems in the databases and report the possible stem-suffix search pairs from the databases. The suffix database can also be used standalone to handle out-of-lexicon words by 'guessing' their suffixes. The Akshar Bharati group has been working on the development of morphological analysers for various Indian languages using the Word and Paradigm model, which is now extended to Sanskrit as well. The complexity of Sanksrit morphological analyser being more when compared to other Indian languages, and at the same time availability of various databases of readymade word forms, the group followed a different approach taking leverage of available resources and data bases for Sanskrit. We describe this approach here.

4 Building a wide coverage morphological analyser quickly

Developing a morphological analysers by designing a generator and then using a Finite State Automata to analyse is a natural choice. But in case of Sanskrit, writing a morphological generator with a reasonable coverage itself is a few man years effort. At the same time, Sanskrit is rich in various databases. For example, Dhāturatnākara is a list of finite verb forms for various verbs for different lakāras (tense-mood-aspect). In a strict sense it does not follow the Pāṇinian Dhātupāṭha, and at some places also deviates a bit from Pāṇinian grammar, but is still a good starting point. Kṛdanta-rūpamālā gives the list of frequently occuring non-finite verbal forms for various verbs. Rūpacandrikā lists all the noun paradigms. These books are being used by people with some exposure of Sanskrit Grammar as an aid to understand and interpret various Sanskrit texts. As such these time-tested books serve as a rich source of readymade databases for developing a reasonable coverage Sanskrit morphological analyser quickly.

Morphological analysers being an important module needed in any NLP application, we decided to take leverage of these existing resources and built a system quickly. The existing resources provide all the analysed forms, from which a Finite State Transducer is built. There are two advantages of this approach. The first one is – in a short span of time, a system with reasonable coverage is available for use which can be plugged in into any other NLP application. The other advantage is, since this system is built from the actual forms listed in the books, and not generated by any software, the data used may be treated as a GOLD standard data for testing the generators built following various approches.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

We used the lt-toolbox[14] – a tool developed by the Apertium group for developing the FST for Sanskrit Analysis. Separate FSTs are built for analysing the basic inflectional forms, the primary derivatives (i.e. kṛadantas) along with the inflected forms that can be generated from them, the secondary derivatives (i.e. taddhitas) along with their inflected forms, compounding forms and finally a FST to recognise the heads of the exo-centric compounds that have gender different from their default gender. These separate FSTs can always be combined together to get a single FST. The trade-off is between the compile time and the run time. The total forms - inflectional, derivational and compounding - our analyser could recognise are 140 million. It is now necessary to evaluate the morphological analyser and also to compare it with other existing morphological analysers to verify the claim that the data used to build the analyser forms a GOLD data.

5 Issues

Initial level of comparison of the noun morphological analysers developed by Scharf-Hyman, Huet and ours brought out some issues.

1. Treatement of feminine adjectives:
   According to the Pāṇinian system at morphological level, nouns and adjectives behave alike. In case of adjectives before forming their feminine forms a feminine suffix is added to the nominal base. Thus *pūrva* is the nominal base in masculine and neuter, and pūrvā in feminine. However, Pāṇini does not treat the nominal base with feminine suffix as a *prātipadika*[15]. Now the question is should the analysis of feminine forms produce the root as pūrvā or pūrva. From semantics point of view, it is reasonable to agree that pūrva is the base in all the three genders. It brings in brevity, and also the organisation of the dictionary is simple.
2. Treatment of homonymy:
   Another issue is related to homonymy. The question is should the morphologically indistinguishable forms with the same stem but different meanings be analysed seperately? This in fact is difficult to handle as it is very difficult to have consensus on whether a word is homonymous or polysemous. It would be desirable, then to distinguish the roots at morphological level, provided they have different behaviour at this level. For example, there are two different seventh case singular forms of the word *sva* viz. *sve* and *svasmin*. These two forms in fact relate to two different roots. In one case *sva* means the 'self', whereas in other case it means 'one's own property'.
3. Level of analysis:
   There is always an issue in the representation of analysis in case of derivational morphology. For example, for the word *gacchati*, should the machine produce the output as seventh case singular form of the nominal base *gacchat*

[14] http://www.apertium.org
[15] ṅyāp-prātipadikāt(4.1.2)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

or should it further add the information that the word *gacchat* is derived from the verbal root *gam* by adding a *kṛt* suffix? With the use of computers, however, this is not an issue, since computers provide several ways of presenting the information in layers, thereby hiding the unnecessary information. While designing a tool for comparison of analysis generated by various analysers, one needs to take this factor into account.

Many more such issues will surface, once we extend our comparison for verbal inflections and other forms.

5.1 Evaluation Parameters

Typically the evaluation metrics of any tool involves two parameters – Precision and Recall. Precision tells you the confidence which you can have on the performance of the tool, and the Recall gives you the coverage.

Since the morphological analyser gives more than one answers, the evaluation of Precision and Recall should range over all possible analysis and not over the words. Further only these two parameters for evaluation of morphological analyser are not enough. We need a more sophisticated measure which tells us which of the answers produced by the morphological analyser are wrong, and also how many answers are missing.

5.2 Precision and Recall

Let the total number of words be N.
Let A_i denote number of possible analysis for the word W_i.
Thus total number of possible answers for all the words together is
$T = \sum_{i=1}^{n} A_i$.
Let B_i denote total answers produced by the morphological analyser for a word W_i.
Let C_i denote the number of correct answers (true positives) and
D_i denote the number of wrong answers (false positives).
Thus $B_i = C_i + D_i$.

Sum of all the correct answers is $C = \sum_{i=1}^{n} C_i$.
Sum of all the wrong answers is $D = \sum_{i=1}^{n} D_i$.
Precision $P = C/(C + D)$
Recall $R = C/T$

5.3 6 point evaluation

Since the morphological analyser produces more than one answers, it will be appropriate to carry out more detailed evaluation of the morpholoical analyser than just evaluating the precision and recall values. Precision-Recall gives general

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

overall impression about the performance of a system. A more detailed evaluation is necessary to know what kind of words are over analysed, which are under analysed, etc. There are 3 parameters viz. an analysis produced by the machine is either correct (true positive) or wrong (false positive) or machine has missed (false negative) an analysis. So when a machine produces an output, it may have zero or more correct analysis, zero or more wrong analysis and zero or more missing analysis. Accordingly, we have the following possibilities:

sr no	Correct (True positives)	Missing (False Negatives)	Wrong (False Positives)	remark
1	1	1	1	ativyāpti as well as avyāpti
2	1	1	0	avyāpti
3	1	0	1	ativyāpti
4	1	0	0	Ideal Case
5	0	1	1	sp case of 1
6	0	1	0	Unrecognised words (sp case of 2)
The other two possibilities will never arise				

The frequency count of each of these 6 cases, and also the morphological phenomena related to these cases will help the developers of morphological analysers to decide which aspect of morphology needs further attention for improvement.

5.4 Evaluation Methodology

Several Sanskrit texts tagged maually at various levels – at the level of sandhi splits (*padaccheda*), morphological analysis (*pada viśleṣaṇa*), sentential analysis (*anvaya*, *śābdabodha*), etc. are available[16]. These manually tagged data may be used for evaluating the system performance. However, comparing the machine produced morphological analysis with these manually tagged texts will give the Precision Recall values only for the 'words in a given context'. These values will be biased because it is possible that the word has some other analysis in some other context which goes un-noticed or un-evaluated. Better evaluation strategy would be to prepare a GOLD standard data for evaluation of morphological analysers. This data contains all possible analysis of the words selected carefully to represent various morph phenomenon. The evaluation – either Precision Recall or 6 point may be done against this GOLD standard data. The possible drawback of this method is, it is possible that missing out some rare analysis lowers down the performance of a system. For all practical purposes, it is also necesary to know the performance on more frequent analysis. For this purpose, one may use manually analysed texts.

[16] E.g. Saṁkṣeparāmāyaṇam by Rashtriya Sanskrit Sansthan, Rāmopākhyāna by Peter Scharf, Bhagvad-geeta by Geeta Press, Gorakhpur, to name a few.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

6 Conclusion

In this paper we have presented a pictorial representation of the Sanskrit morphology as described by Pāṇini. Further we discussed in brief how we can take the leverege of various available resources and build the Sanskrit morphological analyser quickly, that can serve as a model to compare and evaluate other morphological analysers. We also suggested two different criteria for the evaluation of morphological analysers. An indicative list of issues involved in the comparison of various analysers based on our experience is given. It is now a high time to carry out such comparison rigourously to smoothen out the differences in the representation if any.

References

1. B. S. Gillon: *Autonomy of word formation: evidence from Classical Sanskrit.* Indian Linguistics, 56 (1-4), pages 1552, 1995.
2. B. S. Gillon: *Exocentric Compounds in Classical Sanskrit.* In G. Huet, A. Kulkarni, and P. Scharf, editors, Sanskrit Computational Linguistics 1 and 2. Springer-Verlag LNAI 5402, 2009.
3. B. S. Gillon: *Tagging Classical Sanskrit Compounds.* In A. Kulkarni and G. Huet, editors, Sanskrit Computational Linguistics 3, pages 98-105, Springer-Verlag, LNAI 5406, 2009.
4. P. Goyal, A. Kulkarni and L. Behra: *Computer Simulation of Aṣṭādhyāyī: Some Insights* In A. Kulkarni, and G. Huet, editors, Sanskrit Computational Linguistics 1 and 2, pages 139-161, Springer-Verlag LNAI 5402, 2009.
5. O. Hellwig: *Sanskrit Tagger: A Stochastic Lexical and POS Tagger for Sanskrit.* In G. Huet, A. Kulkarni, and P. Scharf, editors, Sanskrit Computational Linguistics 1 and 2, pages 266277. Springer-Verlag LNAI 5402, 2009.
6. O. Hellwig: *Extracting dependency trees form from Sanskrit texts*, In A. Kulkarni and G. Huet, editors, Sanskrit Computational Linguistics 3, pages 106115. Springer-Verlag LNAI 5406, 2009.
7. G. Huet: *A functional toolkit for morphological and phonological processing, application to a Sanskrit tagger.* J. Functional Programming, 15,4:573614, 2005.
8. G. Huet: *Formal Structure of Sanskrit Text: Requirement Analysis for a Mechanical Sanskrit Processor.* In G. Huet, A. Kulkarni, and P. Scharf, editors, Sanskrit Computational Linguistics 1 and 2. Springer-Verlag LNAI 5402, 2009.
9. Pandit Ishvarachandra: *Aṣṭādhyāyī.* Chaukhamba Sanskrit Pratisthan, Delhi, 2004.
10. G. N. Jha, M. Agrawal, Subash, S. K. Mishra, D. Mani, D. Mishra, M. Bhadra and S. K. Singh: *Inflectional Morphology Analyzer for Sanskrit* In A. Kulkarni, and G. Huet, editors, Sanskrit Computational Linguistics 1 and 2, pages 219-238, Springer-Verlag LNAI 5402, 2009.
11. A. Kulkarni and V. Sheeba: *Building a wide coverage Morphological Analyser for Sanskrit: A practical approach.* Invited speech at 'First National Symposium on Modeling and Shallow Parsing of Indian Languages', 31st March - 4th April 2006, IIT Mumbai.(http://sanskrit.uohyd.ernet.in/faculty/amba/)
12. Mahavira: *Pāṇini as Grammarian (With special reference to compound formation).* Bharatiya Vidya Prakashan [Delhi - Varanasi], India, June 1978.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

13. A. Mishra: *Modelling the Grammatical Circle of the Pinian System of Sanskrit Grammar* In A. Kulkarni, and G. Huet, editors, Sanskrit Computational Linguistics 3, pages 40-55, Springer-Verlag LNAI 5406, 2009.
14. A. Mishra: *Simulating the Pinian System of Sanskrit Grammar* In A. Kulkarni, and G. Huet, editors, Sanskrit Computational Linguistics 1 and 2, pages 127-138, Springer-Verlag LNAI 5402, 2009.
15. Vasudeva Lakshman Shastri Panasikar: *Siddhānta Kaumudi.* Meharchand Lachamandas Publications, New-Delhi, 1985.
16. Peter M. Scharf: *Levels in Pāṇini's Aṣṭādhyāyī.* In A. Kulkarni, and G. Huet, editors, Sanskrit Computational Linguistics 3, pages 66-77, Springer-Verlag LNAI 5406, 2009.
17. S. Ramasubba Sastri: *Kṛdanta-Rūpamālā*, The Sanskrit Education Society, Madras, 1989
18. S Sridhar and S Varkhedi: *Computational Structure of the Aṣṭādhyāyī and Conflict Resolution Techniques* In A. Kulkarni, and G. Huet, editors, Sanskrit Computational Linguistics 3, pages 56-65, Springer-Verlag LNAI 5406, 2009.
19. Muni Lavannya Vijaya Suri: *Dhātu-Ratnākara* Bharatia Book Corporation, Delhi,2005.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

Sanskrit Compound Processor

Anil Kumar[1], Vipul Mittal[2] and Amba Kulkarni[1]

[1] Department of Sanskrit Studies, University of Hyderabad, India
anil.lalit22@gmail.com, apksh@uohyd.ernet.in
[2] Language Technologies Research Centre, IIIT, Hyderabad, India
vipulmittal@research.iiit.ac.in

Abstract. Sanskrit is very rich in compound formation. Typically a compound does not code the relation between its components explicitly. To understand the meaning of a compound, it is necessary to identify its components, discover the relations between them and finally generate a paraphrase of the compound. In this paper, we discuss the automatic segmentation and type identification of a compound using simple statistics that results from the manually annotated data.

Key words: Sanskrit Compound Splitter, Sanskrit Compound Type Identifier, Sanskrit Compounds, Optimality Theory.

1 Introduction

In recent years Sanskrit Computational Linguistics has gained momentum. There have been several efforts towards developing computational tools for accessing Sanskrit texts ([12], [14], [23], [16], [7], [1]). Most of these tools handle morphological analysis and sandhi splitting. Some of them ([10], [13], [8]) also do the sentential parsing. However, there have been almost negligible efforts in handling Sanskrit compounds, beyond segmentation.

Sanskrit is very rich in compound formation. The compound formation being productive, forms an open-set and as such it is also not possible to list all the compounds in a dictionary. The compound formation involves a mandatory *sandhi*[3]. But mere *sandhi* splitting does not help a reader in identifying the meaning of a compound. Typically a compound does not code the relation between its components explicitly. To understand the meaning of a compound, thus, it is necessary to identify its components, discover the relations between them, and finally produce a *vigrahavākya*[4] of the compound. Gillon [6] suggests tagging of compounds by enriching the context free rules. He does so by specifying the vibhakti, marking the head and also specifying the enriched category

[3] *Sandhi* means euphony transformation of words when they are consecutively pronounced. Typically when a word w_1 is followed by a word w_2, some terminal segment of w_1 merges with some initial segment of w_2 to be replaced by a "smoothed" phonetic interpolation, corresponding to minimising the energy necessary to reconfigure the vocal organs at the juncture between the words. [11]

[4] An expression providing the meaning of a compound is called a *vigrahavākya*.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

of the components. He also points out how certain components such as *'na'* provide a clue for deciding the type of a compound. *Pāṇini* captures special cases and formulates rules to handle them. Implementing these rules automatically is still far from reality on account of the semantic information needed for the implementation. In the absence of such semantic information, statistical methods have proved to be a boon for the language engineers. These statistical tools use manually annotated data for automatic learning, and in turn develop a language model, which is then used for analysis. In what follows we discuss the automatic segmentation and type identification of compounds using simple statistics that result from the manually annotated data.

2 Sanskrit Compounds

The Sanskrit word *samāsaḥ* for a compound means *samasanam* which means "combination of more than one word into one word which conveys the same meaning as that of the collection of the component words together". While combining the components together, a compound undergoes certain operations such as loss of case suffixes, loss of accent, etc.. A Sanskrit compound thus has one or more of the following features ([17], [6]):

1. It is a single word (*ekapadam*).[5]
2. It has a single case suffix (*ekavibhaktikam*) with an exception of *aluk* compounds such as *yudhiṣṭiraḥ*, where there is no deletion of case suffix of the first component.
3. It has a single accent(*ekasvaraḥ*).[6]
4. The order of components in a compound is fixed.
5. No words can be inserted in between the compounds.
6. The compound formation is binary with an exception of *dvandva* and *bahupada bahuvrīhi*.
7. Euphonic change (*sandhi*) is a must in a compound formation.
8. Constituents of a compound may require special gender or number different from their default gender and number. e.g. *pāṇipādam, pācikābhāryaḥ*, etc.

Though compounds of 2 or 3 words are more frequent, compounds involving more than 3 constituent words with some compounds even running through pages is not rare in Sanskrit literature. Here are some examples of Sanskrit compounds involving more than 3 words.

1. वेदवेदाङ्गतत्त्वज्ञः[7]=वेद-वेदाङ्ग-तत्त्व-ज्ञः
2. प्रवरमुकुटमणिमरीचिमञ्जरीचयचर्चितचरणयुगल[8]=प्रवर-मुकुट-मणि-मरीचि-मञ्जरी-चय-चर्चित-चरण-युगल

---

[5] aikapadyam aikasvaryam ca samāsatvāt bhavati [Kāśikā 2.1.46]
[6] aikapadyam aikasvaryam ca samāsatvāt bhavati [Kāśikā 2.1.46]
[7] *Rāmāyaṇam 1-1-14*
[8] *Pañcatantram in kathamukham*

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

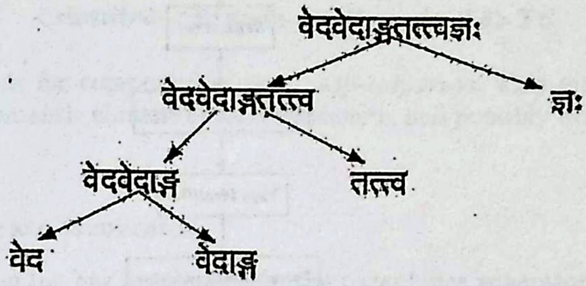


Fig. 1. Constituency representation of वेदवेदाङ्गतत्त्वज्ञः

3. जलादिव्यापकपृथिवीत्वाभावप्रतियोगिपृथिवीत्ववती[9]=जल-आदि-व्यापक-पृथिवीत्व-अभाव-प्रतियोगि-पृथिवीत्ववती

The compounds are formed with two words at a time and hence they can be represented faithfully as a binary tree, as shown in figure 1.

Semantically *Pāṇini* classifies the Sanskrit compounds into four major types:

- Tatpuruṣaḥ: (Endocentric with head typically to the right),
- Bahuvrīhiḥ: (Exocentric),
- Dvandvaḥ: (Copulative), and
- Avyayībhāvaḥ: (Endocentric with head typically to the left and behaves as an indeclinable).

This classification is not sufficient for generating the paraphrase. For example, the paraphrase of a compound *vṛkṣamūlam* is *vṛkṣasya mūlam* and *gṛhagataḥ* is *gṛham gataḥ*, though both of them belong to the same class of *tatpuruṣaḥ*. Based on their paraphrase, these compounds are further sub-classified into 55 sub-types. Appendix-A provides the classification and Appendix-B describes the paraphrase associated with each tag.

3 Compound Processor

Understanding a compound involves (ref Fig 2)

1. Segmentation (समासपदच्छेदः),
2. Constituency Parsing (समासपदान्वयः),
3. Compound Type Identification (समस्तपदपरिचायकः), and
4. Paraphrasing (विग्रह-वाक्यम्).

These four tasks form the natural modules of a compound processor. The output of one task serves as an input for the next task until the final paraphrase is generated.

[9] *Kevalavyatireki-prakaraṇam : Maṇikaṇa* [22]

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

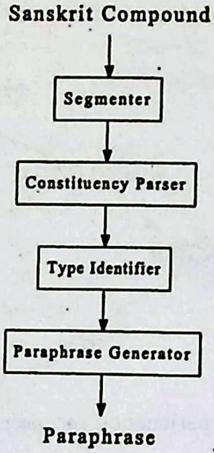


Fig. 2. Compound Analyser

3.1 Segmenter

The task of this module is to split a compound into its constituents. For instance, the compound

sumitrānandavardhanaḥ

is segmented as

sumitrā-ānanda-vardhanaḥ

Each of the constituent component except the last one is typically a compounding form (a bound morpheme)[10].

3.2 Constituency Parser

This module parses the segmented compound syntactically by pairing up the constituents in a certain order two at a time. For instance,

sumitrā-ānanda-vardhanaḥ

is parsed as

<sumitrā-<ānanda-vardhanaḥ>>

3.3 Type Identifier

This module determines the type on the basis of the components involved. For instance,

<sumitrā-<ānanda-vardhanaḥ>>

is tagged as

[10] with an exception of components of an *'aluk'* compound.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

<sumitrā-<ānanda-vardhanaḥ>T6>T6

where T6 stands for compound of type *ṣaṣṭī-tatpuruṣa*. This module needs an access to the semantic content of its constituents, and possibly even to the wider context.

3.4 Paraphrase Generator

Finally after the tag has been assigned, the paraphrase generator [16] generates a paraphrase for the compound. For the above example, the paraphrase is generated as:
ānandasya vardhanaḥ = *ānandavardhanaḥ*,
sumitrāyāḥ ānandavardhanaḥ = *sumitrānandavardhanaḥ*.

4 Compound Segmenter

The task of a segmenter is to split a given sequence of phonemes into a sequence of morphologically valid segments [18]. The compound formation involves a mandatory sandhi. Each sandhi rule is a triple (x, y, z) where y is the last letter of the first primitive, z is the first letter of the second primitive, and x is the letter sequence resulting from the euphonic combination. For analysis, we reverse these rules of sandhi and produce $y + z$ corresponding to a x. Only the sequences that are morphologically valid are selected. The segmentation being non-deterministic, segmenter produces multiple splits. To ensure that the correct output is not deeply buried down the pile of incorrect answers, it is natural to prioritize solutions based on some scores. We follow GENerate-CONstrain-EVALuate paradigm attributed to the Optimality Theory [21] for segmentation. As is true of any linguistic theory, the Optimality Theory basically addresses the issue of generation. Nevertheless, there have been successful attempts ([4], [3]) to reverse the process of generation. It will be really challenging to implement the sandhi rules from *Aṣṭādhyāyī* as CONstraints and then reverse them for splitting. In this attempt however, we simply use the 'cooked' sandhi rules in the form of triplets. We describe below the GENerate-CONstrain-EVALuate cycle of the segmenter and the scoring matrix used for prioritizing the solutions.

4.1 Scoring Matrix

A parallel corpus of Sanskrit text in sandhied and unsandhied form is being developed as a part of the Sanskrit Consortium project in India. The corpus contains texts from various fields ranging from children stories, dramas, purāṇas to Ayurveda texts. From around 100K words of such a parallel corpus, 25K parallel instances of sandhied and unsandhied text were extracted. These examples were used to get the frequency of occurence of various sandhi rules. If no instance

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

of a sandhi rule is found in the corpus, for smoothing, we assign the frequency of 1 to this sandhi rule.

We define the estimated probability of the occurrence of a sandhi rule as follows: Let R_i denote the i^{th} rule with f_{R_i} as the frequency of occurrence in the manually split parallel text. The probability of rule R_i is:

$$P_{R_i} = \frac{f_{R_i}}{\sum_{j=1}^{n} f_{R_j}}$$

where n denotes the total number of sandhi rules found in the corpus.

Let a word be split into a candidate S_j with k constituents as $< c_1, c_2, ..., c_k >$ by applying $k-1$ sandhi rules $< R_1, R_2, ..., R_{k-1} >$ in between the constituents. It should be noted here that the rules $R_1, ..., R_{k-1}$ and the constituents $c_1, ..., c_k$ are interdependent since a different rule sequence will result in a different constituents sequence. The sequence of constituents are constrained by a language model whereas the rules provide a model for splitting. We define two measures each corresponding to the constituents and the rules to assign weights to the possible splits.

Language Model Let the unigram probability of the sequence $< c_1, c_2, ..., c_k >$ be PL_{S_j} defined as:

$$PL_{S_j} = \prod_{x=1}^{k} (P_{c_x})$$

where P_{c_x} is the probability of occurrence of a word c_x in the corpus.

Split Model Let the splitting model PS_{S_j} for the sandhi rules sequence $< R_1, R_2, ..., R_{k-1} >$ be defined as:

$$PS_{S_j} = \prod_{x=1}^{k-1} (P_{R_x})$$

where P_{R_x} is the probability of occurrence of a rule R_x in the corpus.

Therefore, the weight of the split S_j is defined as the product of the language and the split model as:

$$W_{S_j} = \frac{PL_{S_j} * PS_{S_j}}{k}$$

where the factor of k is introduced to give more preference to the split with less number of segments than the one with more segments.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

4.2 Segmentation Algorithm

The approach followed is GENerate-CONstrain-EVALuate. In this approach, all the possible splits of a given string are first generated and the splits that are not validated by the morphological analyser are subsequently pruned out.

Currently we apply only two constraints viz.

- C1 : All the constituents of a split must be valid morphs.
- C2 : All the segments except the last one should be valid compounding forms.

The system flow is presented in Figure 3.

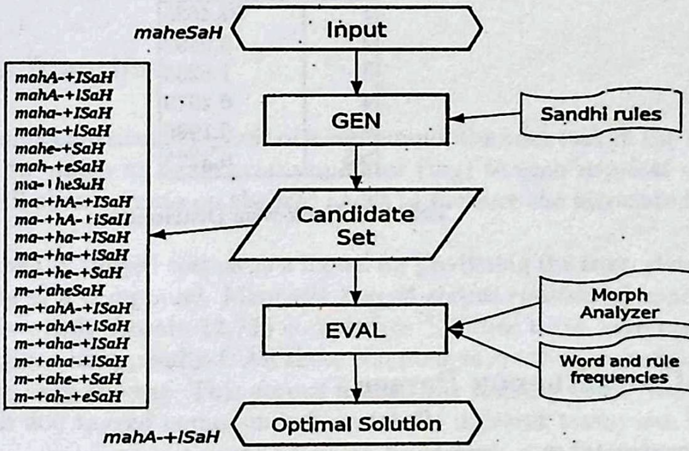


Fig. 3. Compound Splitter: System Data Flow.

The basic outline of the algorithm is:

1. Recursively break a word at every possible position applying a sandhi rule and generate all possible candidates for the input.
2. Pass the constituents of all the candidates through the morph analyser.
3. Declare the candidate as a valid candidate, if all its constituents are recognised by the morphological analyser, and all except the last segment are compounding forms.
4. Assign weights to the accepted candidates and sort them based on the weights as defined in the previous subsection.
5. The optimal solution will be the one with the highest weight.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

Results The current morphological analyser[11] can recognise around 140 million words. Using 2,650 rules and a test data of around 8,260[12] words parallel corpus for testing, we obtained the following results:

– Almost 92.5% of the times, the first segmentation is correct. And in almost 99.1% of the cases, the correct split was among the top 3 possible splits.
– The precision was about 92.46% (measured in terms of the number of words for which first answer is correct w.r.t. the total words for which correct segmentation was obtained).
– The system consumes around 0.04 seconds per string of 15 letters on an average.[13]

The complete rank wise distribution is given in Table 1.

Rank	% of words
1	92.4635
2	5.0492
3	1.6235
4	0.2979
5	0.1936
>5	0.3723

Table 1. Rank-wise Distribution.

5 Constituency Parser

Segmenter takes a compound as an input and produces one or more possible segmentations conditioned by the morphological analyser and the sandhi rules. Constituency parser takes an output of the segmenter and produces a binary tree showing the syntactic composition of the compound corresponding to each of the possible segmentations. Each of these compositions show the possible ways various segments can be grouped. To illustrate various possible parses that result from a single segmentation, consider the segmentation a-b-c of a compound. Since a compound is binary, the three components a-b-c may be grouped in two ways as $< a < bc >>$ or $<< ab > c >$. Only one of the ways of grouping may be correct in a given context as illustrated by the following two examples.

1. < *baddha-* < *jāmbūnada- srajaḥ* >>
2. << *tapas-svādhyāya* >- *niratam* >

[11] available at http://sanskrit.uohyd.ernet.in/ anusaaraka/sanskrit/samsaadhanii/morph/index.html

[12] The test data is extracted from manually split data of *Mahābhāratam*.

[13] Tested on a system with 2.93GHz Core 2 Duo processor and 2GB RAM.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

With 3 components, only these two parses are possible. But as the number of constituents increase, the number of possible ways the constituents can be grouped grows very fast. The constituency parsing is similar to the problem of completely parenthesizing *n+1* factors in all possible ways. Thus the total possible ways of parsing a compound with $n+1$ constituents is equal to a *Catalan number*, C_n [13] where for $n \geq 0$,

$$C_n = \frac{(2n)!}{(n+1)!n!}$$

The task of the constituency parser is then to choose the correct way of grouping the components together, in a given context. To the best of our knowledge no work has been initiated yet that produces the best constituency parse for a given segmentation, in a given context.

6 Type Identifier

After getting a constituency parse of a compound, the next task in the compound analysis is to assign an appropriate operator (tag) to each non-leaf node. This operator will then operate on the leaf nodes to produce the associated meaning.

We use manually tagged corpus as a model for predicting the tags, given a pair of constituents of a compound. Manually tagged corpus consists of approximately 150K words which contain 12,796 compounds[14]. These texts were tagged using the tagset given in appendix-I. All these compounds are thus tagged 'in context' and contain only one tag. This corpus formed the training data. Another small corpus with 400 tagged compounds from totally different texts, was kept aside for testing.

Some features of the manually tagged data

1. Around 10% of the compounds were repeated.
2. The 12,796 tokens of compounds contain 2,630 types of left word and 7,106 types of right word.
3. The frequency distribution of highly frequent tags is shown in Table 2. To study the effect of fine-grained-ness we also merged the sub-types. Table 3 gives the frequency distribution of major tags, after merging the sub-types.

We define,

$P(T/W_1 - W_2)$ = probability that a compound $W_1 - W_2$ has tag T.

Assuming that occurrence of W_1 and W_2 are independent,

[14] Only compounds with 2 components were selected

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

Tag	% of words
T6	28.35
Bs6	12.45
K1	9.63
Tn	8.56
Di	7.23
U	5.73

Table 2. Distribution of Fine-grain-Tags

Tag	% of words
T	52.43
B	18.96
K	12.04
D	8.84
U	5.73

Table 3. Distribution of Coarse-Tags

$$P(T/W_1 - W_2) = P(T/W_1) * P(T/W_2)$$

where $P(T/W_i) = \frac{P(T*W_i)}{P(W_i)}$, i= 1,2.

Since the data is sparse, to account for smoothing, we define, for unseen instances,

$P(T.W_i) = \frac{1}{F}$,
$P(W_i) = \frac{1}{F}$,

where F is the total number of manually tagged compounds.

6.1 Performance Evaluation

The test data of 400 words are tagged 'in context'. While our compound tagger does not see the context, and thus suggests more than one possible tag, and ranks them. Normally a tool is evaluated for its coverage and precision. In our case, the tool always produces tags with weights associated with them. The coverage and precision therefore are evaluated based on the ranks of the correct tags. Table4 shows results of 400 words with a coarse as well as fine grained tagset.

	with 55 tags		with 8 tags	
Rank	no of words	% of words	no of words	% of words
1	252	63.0	291	72.7
2	44	10.9	53	13.2
3	29	7.2	38	9.5

Table 4. Precision of Type Identifier.

Thus, if we consider only the 1^{st} rank, the precision with 8 tags is 72.7% and with 55 tags, it is 63.0%. The precision increases substantially to 95.4% and 81.1% respectively if we take 1^{st} three ranks into account.

The performance of the type identifier can be further improved by using semantically tagged lexicon. There are around 200 rules in the *Aṣṭādhyāyī* which

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

provide semantic contexts for various compounds. One such example is the *Pāṇini*'s sūtra *annena vyañjanam* (2.1.34). This sūtra gives a condition for forming *tṛtīyā tatpuruṣaḥ* compound. Thus what is required is a list of all words that denote eatables. A lexicon rich with such semantic information would enhance the performance of the tagger further. The compound processor with all the functionality described above is available online at

http://sanskrit.uohyd.ernet.in/samAsa/frame.html

References

1. Bharati, A., Kulkarni, A.P., and Sheeba, V.: *Building a wide coverage Sanskrit Morphological Analyser: A practical approach.* In: The First National Symposium on Modelling and Shallow Parsing of Indian Languages, IIT-Bombay. (2006)
2. Bhat, G.M.: *Samāsaḥ.* Samskrita Bharati, Bangalore, Karnataka, 2006.
3. Fortes, F.C.L., and Roxas, R.E.O.: *Optimality Theory in Morphological Analysis* In: National Natural Language Processing Research Symposium, January 2004.
4. Fosler, J.E.: *On Reversing the Generation Process in Optimality Theory.* In : Proceedings of the Association for Computational Linguistics. (1996)
5. Gillon, B.S.: *Exocentric Compounds in Classical Sanskrit.* In: Proceeding of the First International Symposium on Sanskrit Computational Linguistics(SCLS-2007), Paris, France, 2007.
6. Gillon, B.S.: *Tagging Classical Sanskrit Compounds.* In: Sanskrit Computational Linguistics 3, pages 98-105, Springer-Verlag LNAI 5406. (2009)
7. Hellwig, O.: *Sanskrit Tagger: A Stochastic Lexical and POS Tagger for Sanskrit.* In: Sanskrit Computational Linguistics 1 & 2, pages 266-277, Springer-Verlag LNAI 5402. (2007)
8. Hellwig, O.: *Extracting Dependency Trees from Sanskrit Texts.* In: Sanskrit Computational Linguistics 3, pages 106-115, Springer-Verlag LNAI 5406. (2009)
9. Hoeks, J.C.J., and Hendriks, P.: *Optimality Theory and Human Sentence Processing: The Case of Coordination.* In: Proceedings of the 27th Annual Meeting of the Cognitive Science Society, Erlbaum, Mahwah, NJ, pp. 959–964. (2005)
10. Huet, G.: *Shallow syntax analysis in Sanskrit guided by semantic nets constraints.* In: Proceedings of International Workshop on Research Issues in Digital Libraries, Kolkata. (2006)
11. Huet, G.: Lexicon-directed Segmentation and Tagging of Sanskrit. XIIth World Sanskrit Conference, Helsinki, Finland, Aug. 2003. In Themes and Tasks in Old and Middle Indo-Aryan Linguistics, Eds. Bertil Tikkanen and Heinrich Hettrich. Motilal Banarsidass, Delhi, 2006, pp. 307-325.
12. Huet, G.: *Formal structure of Sanskrit text: Requirements analysis for a Mechanical Sanskrit Processor.* In: Sanskrit Computational Linguistics 1 & 2, pages 162-199, Springer-Verlag LNAI 5402. (2007)
13. Huet, G.: *Sanskrit Segmentation.* In: South Asian Languages Analysis Roundtable XXVIII, Denton, Ohio. (October 2009)
14. Jha, G.N. and Mishra, S.K.: *Semantic Processing in Pāṇini's Kāraka System.* In: Sanskrit Computational Linguistics 1 & 2, pages 239-252, Springer-Verlag LNAI 5402. (2007)
15. Kulkarni, A.P., Shukla, D.: *Sanskrit Morphological Analyser: Some Issues.* In: To appear in Bh.K Festschrift volume by LSI. (2009)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

16. Kulkarni, A.P., Kumar, A., Sheeba, V.: *Sanskrit compound paraphrase generator*. In: Proceedings of ICON-2009: 7th International Conference on Natural Language Processing, Macmillan Publishers, India.
17. Mahavira: *Pāṇini as Grammarian (With special reference to compound formation)*. Bharatiya Vidya Prakashan [Delhi - Varanasi], India, June 1978.
18. Mittal, V.: *Automatic Sanskrit Segmentizer using Finite State Transducers*. In: Proceeding of Association for Computational Linguistics - Student Research Workshop. (2010)
19. Murty, M.S.: *Sanskrit Compounds-A Philosophical Study*. Chowkhamba Sanskrit Series Office, Varanasi(India), 1974.
20. Pandit Ishvarachandra: *Aṣṭādhyāyī*. Chaukhamba Sanskrit Pratisthan, Delhi, 2004.
21. Prince, A., Smolensky, P.: *Optimality Theory: Constraint Interaction in Generative Grammar*. In: RuCCS Technical Report 2 at Center for Cognitive Science, Rutgers University, Piscataway. (1993)
22. Sarma, E.R.S.: *Maṇikaṇa : A Navya-Nyāya Manual*. The Adyar Library and research Centre, Madras. (1960)
23. Scharf, P.M.: *Levels in Pāṇini's Aṣṭādhyāyī*. In: Sanskrit Computational Linguistics 3, pages 66-77, Springer-Verlag LNAI 5406. (2009)
24. Yuret, D., Biçici, E.: *Modeling Morphologically Rich Languages Using Split Words and Unstructured Dependencies*. In: ACL-IJCNLP, Singapore. (2009)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

Appendix-A : Sanskrit Compound Type and sub-type Classification

अव्ययीभावः		कर्मधारयः	
Compound sub-types	Tags	Compound sub-types	Tags
अव्यय-पूर्वपदः	A1	विशेषण-पूर्वपद-कर्मधारयः	K1
अव्यय-उत्तरपदः	A2	विशेषण-उत्तरपद-कर्मधारयः	K2
तिष्ठद्गुप्रभृति	A3	विशेषण-उभयपद-कर्मधारयः	K3
संख्यापूर्वपद-नद्युत्तरपद	A4	उपमान-पूर्व-पद-कर्मधारयः	K4
नद्युत्तरपद-अन्यपदार्थे संज्ञायाम्	A5	उपमान-उत्तर-पद-कर्मधारयः	K5
संख्यापूर्वपद-वंश्योत्तरपदः	A6	अवधारणापूर्वपदः-कर्मधारयः	K6
पारे-मध्ये-पूर्वपद षष्ठ्युत्तरपद	A7	सम्भावनापूर्वपद-कर्मधारयः	K7
		मध्यमपदलोपी-कर्मधारयः	Km
तत्पुरुषः		बहुव्रीहिः	
Compound sub-types	Tags	Compound sub-types	Tags
प्रथमातत्पुरुषः	T1		
द्वितीयातत्पुरुषः	T2	द्वितीयार्थ-बहुव्रीहिः(समानाधिकरणः)	Bs2
तृतीयातत्पुरुषः	T3	तृतीयार्थ-बहुव्रीहिः(समानाधिकरणः)	Bs3
चतुर्थीतत्पुरुषः	T4	चतुर्थ्यर्थ-बहुव्रीहिः(समानाधिकरणः)	Bs4
पञ्चमीतत्पुरुषः	T5	पञ्चम्यर्थ-बहुव्रीहिः(समानाधिकरणः)	Bs5
षष्ठीतत्पुरुषः	T6	षष्ठ्यर्थ-बहुव्रीहिः(समानाधिकरणः)	Bs6
सप्तमीतत्पुरुषः	T7	सप्तम्यर्थ-बहुव्रीहिः(समानाधिकरणः)	Bs7
नञ्तत्पुरुषः	Tn	दिग्वाचक-बहुव्रीहिः(समानाधिकरणः)	Bsd
समाहार-द्विगुः	Tds	प्रहरणविषयक-बहुव्रीहिः(समानाधिकरणः)	Bsp
तद्धितार्थद्विगुः	Tdt	ग्रहणविषयक-बहुव्रीहिः(समानाधिकरणः)	Bsg
उत्तरपदद्विगुः	Tdu	अस्त्यर्थ-मध्यमपदलोपी(नञ्)-बहुव्रीहिः	Bsmn
गतिसमासः	Tg	प्रादि-बहुव्रीहिः	Bvp
कुसमासः	Tk	संख्योभयपद-बहुव्रीहिः(समानाधिकरणः)	Bss
प्रादिसमासः	Tp	उपमानपूर्वपद-बहुव्रीहिः(समानाधिकरणः)	Bsu
मयूरव्यंसकादिः	Tm	व्यधिकरण-बहुव्रीहिः	Bv
तत्पुरुषः बहुपदः	Tb	सङ्ख्योत्तरपदः व्यधिकरण-बहुव्रीहिः	Bvs
तत्पुरुषः उपपदः	U	सहपूर्वपद-व्यधिकरण-बहुव्रीहिः	BvS
		उपमानपूर्वपद-व्यधिकरण-बहुव्रीहिः	BvU
		बहुपदः-बहुव्रीहिः	Bb
द्वन्द्वः		केवल	
Compound type:	Tags	Compound type:	Tags
इतरेतरयोग-द्वन्द्वः	Di	केवल समासः	S
समाहार-द्वन्द्वः	Ds	द्विरुक्तिः	
एकशेष	E	द्विरुक्तिः	d

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

Appendix-B : Rules for Paraphrase Generation.

	अव्ययीभावः
1	<x-y>A1 => y{6} f{x} where f maps x to the noun with same semantic content. A function f needs to be defined.
2	<x-y>A2 => x{3} विपरीतम् वृत्तम्
3	<x-y>A4 => x{6} y{6} समाहारः
4	<x-y>A5 => x'{1} y{1} यस्मिन् देशे x' has same gender as that of y'.
5	<x-y>A6 => x{6} y{6} समाहारः if x = द्वि, both x and y will be in द्विवचनम्
6	<x-y>A7 => x{6} y
	तत्पुरुषः
7	<x-y> T_n => x{n} y $2 \leq n \leq 7$
8	<x-y>Tn => x{n} y
9	<x-y>Tds => x{6;ba} y{6;ba} समाहारः
10	<x-y-z>Tb = x{1} y{1} z{1}
	कर्मधारयः
11	<x-y>K1 => x{1} तत् y{1} च
12	<x-y>K2 => x{1} च y{1} च
13	<x-y>K3 => x{1} च असौ y{1} च
14	<x-y>K4 => x{1} इव y{1}
15	<x-y>K5 => x{1} y{1} इव
16	<x-y>K6 => x{1} एव y{1}
17	<x-y>K7 => x{1} इति y{1}
	बहुव्रीहिः
19	<x-y>Bsd => x{6} च y{6} च यदन्तरालम्
20	<x-y>Bsp => x{3} च y{3} च प्रहृत्य इदम् युद्धम् प्रवृत्तम्
21	<x-y>Bsg => x{7}-y{7}+ गृहीत्वा इदम् युद्धम् प्रवृत्तम्
22	<x-y>Bsmn => x'-y{1} यस्य
23	<x-y>Bss = > x{1} वा y{1} यस्य
24	<x-y>Bsu => x{1} इव y{1} यस्याः
25	<x-y>Bv => x y{1} यस्य
26	<x-y>Bvs => y{6} x' ये सन्ति ते
27	<x-y>BvS => y{3} सह
28	<x-y>BvU => x{6} इव y यस्य
	द्वन्द्वः
29	<x-y>Di => x{1} च y{1} च
30	<x-y>Ds => x{1} च y{1} च
31	<x-y>S => y{1} x{1}
32	<x-y>d => x y

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

संस्कृत-समास-प्रक्रिया

अनिल-कुमारः[1], विपुल-मित्तलः[2] अम्बा-कुलकर्णी[1]

[1]संस्कृताध्ययनविभागः, हैदराबादविश्वविद्यालयः, हैदराबाद, भारतम्

anil.lalit22@gmail.com, apksh@uohyd.ernet.in

[2]Language Technology Research Centre, IIIT, Hyderabad, India.

vipulmittal@research.iiit.ac.in

सारांशः

संस्कृतभाषायां समासप्रयोगानामाधिक्यं दृश्यते। तत्र समस्तपददर्शनेनैव समासे प्रयुक्ततत्तत्पदस्य सम्बन्धः स्पष्टरूपेण न प्रतीयते। तस्मात् समासार्थप्राप्त्यर्थं प्रथमं समस्तपदे विद्यमानपदेषु परस्परान्वयः ज्ञातव्यः। एवञ्च तद्विग्रहवाक्यमपि प्राप्तव्यम्। प्रकृतसमुद्गतचर्चायां समाससन्दर्भे गणितशास्त्रे उपयुज्यमानसांख्यिकिप्रक्रियासाहाय्येन यन्त्रेण समासच्छेदः(समासे विद्यमानपदानां पृथग्दर्शनम्) समासभेदनिर्धारणञ्च कथमिति पत्रेऽस्मिन् प्रस्तूयते।

1. परिचयः

प्रगतवर्षेषु संस्कृतसङ्गणकभाषाविज्ञानक्षेत्रे बहुविधकार्याणि जातानि। तत्र बहुभिः विद्वद्भिः कृतपरिश्रमैः महता प्रयासेन संस्कृतज्ञानार्थं ([12],[14],[23],[16],[7],[1]) नानाविधसङ्गणकीयोपकरणानि निर्मितानि। संस्कृतपदविश्लेषकः, सन्धिसंश्लेषकः, सन्धिविच्छेदकः इत्याद्युपकरणानि अधिकतया दृश्यन्ते। तत्र कारकाणामुपर्यपि केचन([10],[13],[8]) कार्यमकुर्वन्। समासः इति विषयम् अवलम्ब्य तु समस्तपदविच्छेदपर्यन्तमेव कार्यं दृश्यते।

संस्कृतवाङ्मये समस्तप्रयोगानाम् आधिपत्यं दरीदृश्यते। समासाः प्रयोगदशायां सरलाः, तदर्थं ते असीमिताः। तथा च न सम्भवति सर्वेषां समस्तपदानां शब्दकोशे योजनं तत्प्रयोगाधिक्यात् तद्बानार्थकत्वाच्च। समाससंरचनायां सन्धिः[1] नित्यः। समासपदे सन्धिविच्छेदेनैव तत्पदार्थः नोपलभ्यते, तस्मिन्विद्यमानपदानां परस्परान्वयाभावात्। समस्तपदार्थावबोधनाय सर्वप्रथमं समस्तपदे विद्यमानपदानां परस्परान्वयः ज्ञातव्यः, तस्मात् विग्रहवाक्यं[2] स्वयमेवोपलभ्यते। 'कोन्टेक्स्ट फ्री ग्रामर्' (Context free grammar) इति आधारीकृत्य समासेषु कथं विभक्तिरित्यादिविशेषाः समीचीनरीत्या अनुबन्धितव्याः इति गिल्लन् (Gillon) [6] महोदयः स्पष्टयति। सः समासपदेषु विभक्तीनां, तत्रविद्यमानविशेष्यम्, समासस्थघटकानाञ्च

1 संहितैकपदे नित्या नित्याधातूपसर्गयोः। नित्या समासे वाक्ये सा विवक्षामपेक्षते।।

2 समासस्य यः अर्थः तमेव विग्रहवाक्यम् इति उच्यते।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

अनुबन्धयति। स च सूचयति यत् कुत्रचित् समासघटकैः यथा 'न' इत्यनेन समासघटकेनैव तत्तत्समासभेदस्य परिचयनम् कथम् इति। इदं हि दर्शयति भगवान् पाणिनेः सूक्ष्मबुद्दिमत्त्वम्। तेन एतादृशांशा अपि सूत्रिताः। यन्त्रस्य सन्दर्भज्ञानं न विद्यते इति हेतोः एतत्सर्वस्यापि यान्त्रिकीकरणमधुनापि दूरं स्थीयते। परन्तु सन्दर्भं विनापि सांख्यिकिप्रक्रियामुपयुज्य तत् साधयितुं शक्यते इत्यङ्गीकुर्वन्ति सङ्गणक-भाषावैज्ञानिकाः। सांख्यिक्युपकरणानि मानवीयानुबन्धिकृतोदाहरणानि अधिकृत्य (manaully tagged data) यन्त्रं शिक्षयन्ति। अनन्तरमेतदेव शिक्षणं भाषायाः अध्ययनार्थमुपयुज्यते।

एवमेव समासस्य मानवीयानुबन्धिकृतोदाहरणानि स्वीकृत्य सामान्यसांख्यिकिप्रक्रियामुपयुज्य कथं समासपदच्छेदः (समासे विद्यमानपदानां पृथग्दर्शनम्) समासभेदनिर्धारणञ्च यन्त्रेण कर्तुं शक्यत इत्येतत् विषयं स्वीकृत्य चर्चा अग्रे भविष्यति।

2. संस्कृतसमासः

कस्तावत् समास इति पदार्थः इति विचिकित्सायां 'समसनमिति समासः' इति प्रोक्तम्। बहुपदानां एकपदीकरणं, समस्तपदे यः अर्थः सैव विग्रहवाक्ये भवितव्यम् इति समासपदेनोच्यते। समासप्रक्रियायां विभक्तिलोपादिकार्याणि विधीयन्ते तेषां विवरणञ्च अधः विवृण्यते।

1) समासः सर्वदा ऐकपदं[3] स्यात् ।

2) समासः ऐकविभक्तिकं[4] स्यात्। अलुक्समासे तु अपवादरूपेण पूर्वपदे विभक्तिलोपः न भवति यथा युधिष्ठिरः इत्यादौ युधि इत्यत्र न विभक्तिलोपः।

3) समासः ऐकस्वर्यं[5] स्यात्।

4) समासे विद्यमानपदानां क्रमपरिवर्तनं न स्यात्।[6]

5) समासे व्यवधानं न स्यात्।[7]

6) द्वन्द्वसमासः बहुपदबहुव्रीहिसमासश्च विहाय अन्यसमासेषु द्वयोर्पदयोरेव समासः विधीयते।

7) नित्यः सन्धिः समासेषु।

8) समासे विद्यमानपदानां स्वलिङ्गं स्ववचनञ्च विहाय पृथग्लिङ्गं पृथग्वचनञ्च भवितुम् अर्हति। यथा - पाणिपादम्, पाचिकाभार्यः इति।

संस्कृतवाङ्मये द्विपदात्मक-त्रिपदात्मकसमासप्रयोगाः बहुधा दृश्यन्ते। अपि च बहुषु स्थलेषु बहुपदात्मकसमस्तप्रयोगाः विरलाः न, संस्कृतभाषायाम्। अधः प्रस्तूयन्ते कानिचन् उदाहरणानि ।

3 ऐकपद्यं ऐकस्वर्यं च समासवत् भवति [काशिका – 2.1.46], समर्थः पदविधिः [महाभाष्यम् – 2.1.1]

4 ऐकपद्यं ऐकस्वर्यं च समासवत् भवति [काशिका – 2.1.46], समर्थः पदविधिः [महाभाष्यम् – 2.1.1]

5 समर्थः पदविधिः [महाभाष्यम् – 2.1.1] इति सूत्रे वर्णितम्।

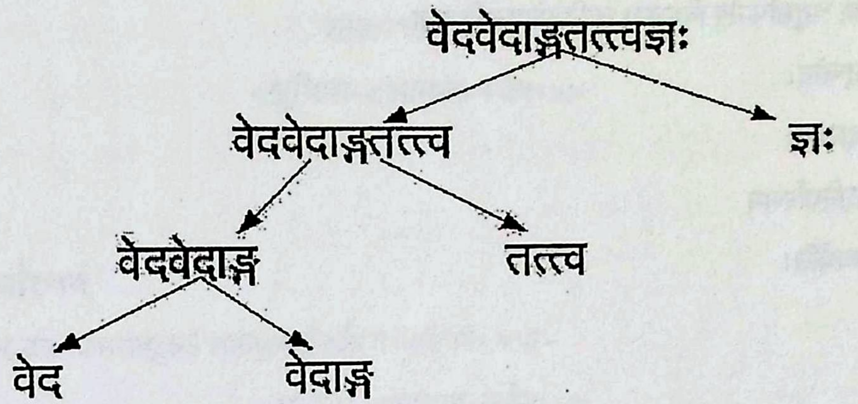
6 समर्थः पदविधिः [महाभाष्यम् – 2.1.1] इति सूत्रे वर्णितम्।

7 समर्थः पदविधिः [महाभाष्यम् – 2.1.1] इति सूत्रे वर्णितम्।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

1) वेदवेदाङ्गतत्त्वज्ञः[8] = वेद-वेदाङ्ग-तत्त्व-ज्ञः



चित्रम् 1 - वेदवेदाङ्गतत्त्वज्ञः इत्यस्य पदान्वयप्रदर्शनम्

बहुपदसमासप्रयोगे द्विपदात्मकसमासप्रयोगानां समावेशः प्रकृतचित्रे -1 द्रष्टुं शक्यते।

2) प्रवरमुकुटमणिमरीचिमञ्जरीचयचर्चितचरणयुगलम्[9] = प्रवर-मुकुट-मणि-मरीचि-मञ्जरी-चय-चर्चित-चरण-युगलम्।

3) जलादिव्यापकपृथिवीत्वाभावप्रतियोगिपृथिवीत्ववती[10] = जल-आदि-व्यापक-पृथिवीत्व-अभाव-प्रतियोगि-पृथिवीत्ववती।

आर्थिकदृष्ट्या भगवता पाणिनिना समासः चतुर्विधः प्रोक्तः। यथा -

1) तत्पुरुषः (प्रायेण उत्तपदार्थप्रधानः)

2) बहुव्रीहिः (प्रायेण अन्यपदार्थप्रधानः)

3) द्वन्द्वः (प्रायेण उभयपदार्थप्रधानः)

4) अव्ययीभावः (प्रायेण पूर्वपदार्थप्रधानः)

एते भेदाः सर्वेषां समस्तपदानां विग्रहवाक्यं निर्मातुं न पर्याप्ताः। यथा 'वृक्षमूलम्' इत्यस्य विग्रहवाक्यं 'वृक्षस्य मूलम्' इति एवञ्च 'गृहगतः' इत्यस्य विग्रहवाक्यं 'गृहं गतः' इति च स्यात्। एतदुभयमपि तत्पुरुषसमासस्योदाहरणम्। परन्तु उभयोरपि विग्रहवाक्यं दृष्ट्वा ज्ञायते यत् षष्ठीतत्पुरुषः, द्वितीयातत्पुरुषः इत्यादिरूपेण उपरितनप्रोक्तप्रतिसमासस्य अन्यभेदाः अपि विद्यन्ते। पृथक्-पृथक् समासानां विग्रहवाक्यानि दृष्ट्वा अस्माभिः 55 समासस्य अन्यभेदाः प्राप्ताः। समासस्य अन्यभेदान् द्रष्टुं कृपया परिशिष्टः दृश्यताम्। परिशिष्टः-अ मध्ये समासस्य इतरभेदानां विवरणं विद्यते। परिशिष्टः-ब मध्ये विग्रहवाक्यनियमाः विद्यन्ते।

8 रामायणम् – 1.1.14

9 पञ्चतन्त्रे कथामुखे प्रोक्तम्।

10 केवलव्यतिरेकी–प्रकरणम् : मणिकणः [22]

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

३. समासप्रक्रिया

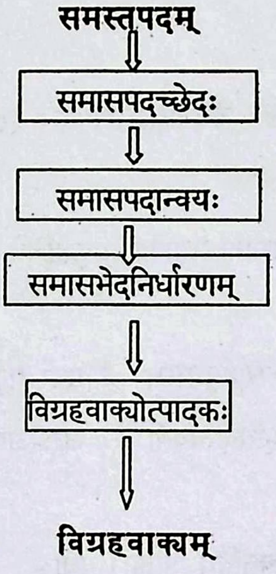
समासप्रक्रिया चतुर्सोपानेषु विभक्ता। तानि सोपानानि यथा -

1) समासपदच्छेदः

2) समासपदान्वयः

3) समासभेदनिर्धारणम्

4) विग्रहवाक्यश्चेति।



3.1 समासपदच्छेदः

सोपानेऽस्मिन् समासे विद्यमानपदानां पृथक्करणं विधीयते। यथा -

सुमित्रानन्दवर्धनः

इत्यस्य पृथक्करणं

सुमित्रा-आनन्द-वर्धनः

इति।

3.2 समासपदान्वयः

समासपदच्छेदानन्तरं ततः प्राप्तपदेषु परस्परान्वयः विधीयते । तथा च द्वयोर्द्वयोः पदयोः समूहीकरणञ्च सोपानेऽस्मिन् विधीयते। यथा -

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

सुमित्रा-आनन्द-वर्धनः

इत्यस्य समूहीकरणं

<सुमित्रा-<आनन्द-वर्धनः>>

इति।

3.3 समासभेदनिर्धारणम्

समासपदान्वयानन्तरं ततः प्राप्तसमूहानां समासभेदनिर्धारणं विधीयते। यथा -

<सुमित्रा-<आनन्द-वर्धनः>>

इत्यस्य समासनामनिर्धारणं

<सुमित्रा-<आनन्द-वर्धनः>T6>T6

इति। अत्र **'T6'** इति षष्ठीतत्पुरुषसमासः इति सूचयति। एतत्सोपानाय आर्थिकपक्षस्यापि अपेक्षा विद्यते।

3.4 विग्रहवाक्यम्

समासभेदनिर्धारणानन्तरं ततः एकैकं समूहम् अवलम्ब्य विग्रहवाक्यम् उत्पद्यते एतस्मिन्सोपाने। यथा -

आनन्दस्य वर्धनः = **आनन्दवर्धनः**

सुमित्रायाः आनन्दवर्धनः = **सुमित्रानन्दवर्धनः**

इति।

4. समासपदच्छेदः

'एकं समस्तपदं स्वीकृत्य तत्रस्थपदानां पृथग्दर्शनमेव समासपदच्छेदः' इत्युपकरणस्य कार्यम्। समासेषु सन्धिः नित्यः। एवञ्च समासच्छेदार्थं सन्धिविच्छेदः प्रथमं सोपानं भवति। सन्धिविच्छेदार्थं सन्धिनियमाः अपेक्ष्यन्ते। सन्धिनियमेषु अस्माभिः पक्षत्रयं स्वीकृतम् । तत्र प्रथमपक्षे सन्धिवर्णानां युगलं स्यात् यथा - रामश्चिनोति इत्यत्र **'श्च'** इति वर्णद्वययुगलम्। द्वितीयपक्षे प्रथमशब्दस्यान्तिमाक्षरं स्यात् यथा - रामश्चिनोति इत्यत्र प्रथमं पदं **'रामस्'** इत्यस्यान्तिमाक्षरं **'स्'** इति। तृतीयपक्षे द्वितीयशब्दस्यादिममक्षरं स्यात् यथा - रामश्चिनोति इत्यत्र द्वितीयं पदं **'चिनोति'** इत्यस्यादिममक्षरं **'च'** इति । सर्वेऽपि सन्धिनियमाः 'प्रथमपक्षम्, द्वितीयपक्षम्, तृतीयपक्षम्' इत्यादिरूपेण समानान्तरेण व्यवस्थिताः। अधुना सन्धिविच्छेदनार्थं प्रथमपक्षमाश्रित्य **'द्वितीयपक्षम् -+ तृतीयपक्षम्'** इति रूपेण विभज्यते यथा - **'श्च'** इत्यस्य **'स्-+च'** इति रूपेण विभाजनम् । एवम् अनुसरणेन तत्र बहुविकल्पानि सन्धिविच्छेदरूपाणि अपेक्षितानपेक्षितरूपाणि च व्युत्पाद्यन्ते। तस्मात् सांख्यिकिप्रक्रियासाहाय्येन अपेक्षितसन्धिविच्छेदितरूपं स्वीक्रियते। एतादृशीं प्रक्रियां भाषावैज्ञानिकाः 'ओप्टीमलिटि थ्योरी (Optimality Theory)' इति नाम्ना व्यवह्रियन्ते। एतत्प्रक्रियायां सोपानत्रयम् आश्रित्य कार्यं विधीयते। सोपानत्रयं नाम 'व्युत्पादनम् - अवरोधनम् - मूल्याङ्कनम् (GENerate – CONstrain – EVALuate)' इति। एतत्सोपानत्रयम्

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

अस्माभिरपि अनुसर्यते समस्तपदच्छेदनार्थम् । वस्तुतः एतस्याः प्रक्रियायाः उपयोगः व्युत्पादनार्थम् (Generation) उपयुज्यते। केचन([4],[3]) एतस्याः प्रक्रियायाः उपयोगः विपरीतदृष्ट्या अपि अकुर्वन् अपि च सफलताम् अप्राप्नुवन्।

अष्टाध्यायीग्रन्थस्थसन्धिविधायकसूत्राणाम् उपयोगः अनपेक्षितरूपान् अवरोद्धुं अर्थात् अवरोधनम् (CONstrain) इति सोपानरूपेण क्रियते। तथा च तेषामेव सन्धिविधाकयकसूत्राणाम् उपयोगः सन्धिविच्छेदनार्थम् अपि करणीयश्चेत् तदेकं जटिलं कार्यं स्यात्। यतोहि सन्धिविधायकसूत्राणि तु केवलं सन्धिः कुत्र आचरणीयः इत्येव सूचयन्ति, न तु सन्धियुक्तपदं कथं विच्छेदनीयम् इति। 'ओप्टीमलिटि थ्योरि (Optimality theory)' इत्यस्याः प्रक्रियायाः सोपानानि अधः विस्तृतरूपेण विवृण्यन्ते।

4.1 स्कोरिंग मैट्रिक्स (Scoring Matrix)

सन्धिसहित-सन्धिरहितोदाहरणानि अस्माभिः भारतसर्वकारस्य 'संस्कृत-समूह-परियोजना (Sanskrit consortium project)' इत्यस्मात् सङ्गृहीतानि। तानि उदाहरणानि विभिन्नग्रन्थेभ्यः स्वीकृतानि। ग्रन्थेषु बालबोधिका, बालकथा, नाटकानि, पुराणानि, आयुर्वेदग्रन्थाः प्रमुखाः सन्ति। एतेभ्यः ग्रन्थेभ्यः शतसहस्रशब्दाः स्वीकृताः। तेषु पञ्चविंशतिसहस्रात्मकानि सन्धियुक्तोदाहरणानां सन्धिविच्छेदनं कारयित्वा समानन्तररूपेण व्यवस्थितानि। एतेषाम् उदाहरणानां सन्धिनियमानाम् आवृत्तिसङ्ख्यां प्राप्तुम् उपयोगः कृतः। यथा सर्वणसन्धिविधायकोदहारणानि कति, कियद्वारञ्च आगतानि इत्यादिरूपेण सन्धिनियमाः व्यवस्थिताः । यदि अपेक्षितसन्धिनियमः न प्राप्यते तर्हि तस्य सामान्यरूपेण 1 इति आवृत्तिसङ्ख्यां विचिन्त्य कार्यं सम्पादयामः इति सिद्धान्तः।

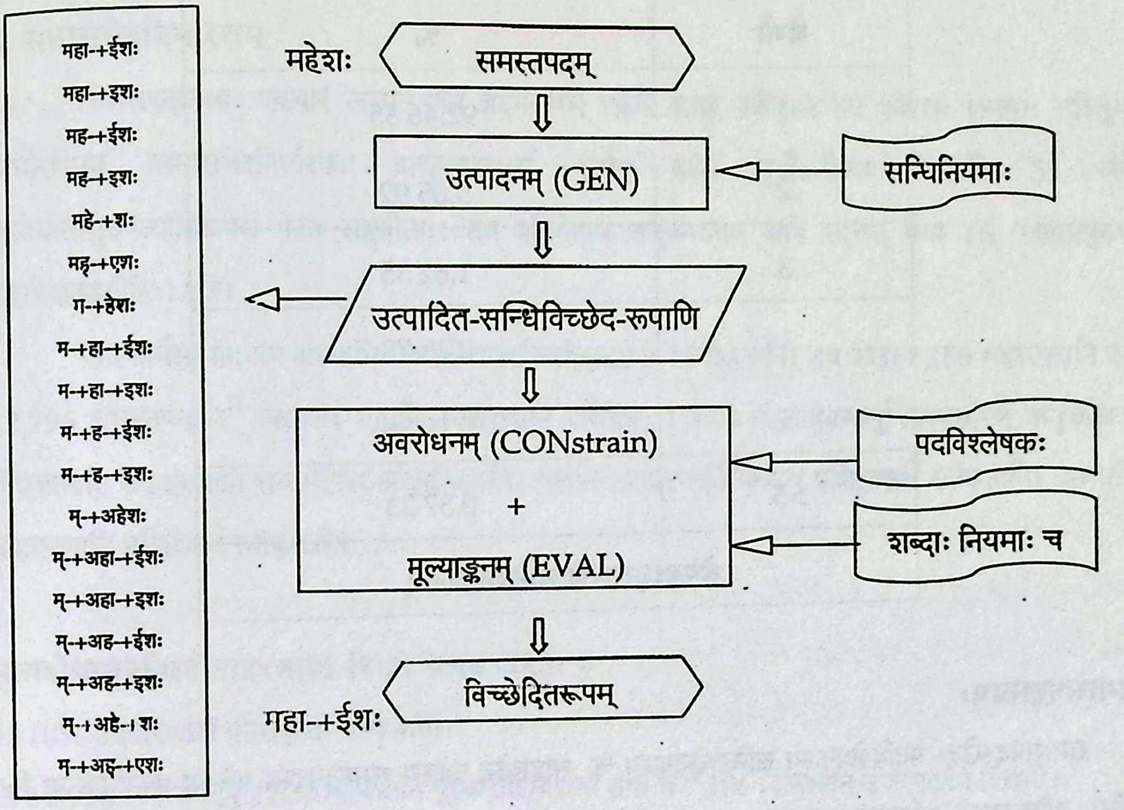
4.2 समासपदच्छेदप्रक्रिया सोपानम् च

यथा पूर्वमेव विचारितं यत् समासपदच्छेदार्थम् अस्माभिः 'ओप्टीमलिटि थ्योरि' (Optimality Theory) इत्यस्याः प्रक्रियायां उपयोगः क्रियते। तथा च तत्र सोपानत्रयम् '**उत्पादनम् - अवरोधनम् - मूल्याङ्कनम्**' इत्येतत् आधारीकृत्य कार्यं विधीयते। उत्पादनम् इत्यस्मिन् सोपाने यावच्छक्यं तावत् अपेक्षित-अनपेक्षित-समासविच्छेदरूपाणाम् उत्पादनं भवति। तदनन्तरं 'पदविश्लेषकः' इत्युपकरणस्य साहाय्येन अनपेक्षितसमासविच्छेदरूपाणि निष्कास्यन्ते।

अवरोधनम् -मूल्याङ्कनम् च इत्यनयोः सोपानयोः अनपेक्षितसमासविच्छेदितरूपाणां निष्कासनम् अवरोधनञ्च भवति। तत्र अवरोधनं निष्कासनञ्च द्विविधा क्रियते। तद्यथा - 1) पदविश्लेषकः इत्युपकरणम् उपयुज्य समस्तपूर्वपद-उत्तरपदयोः साधुत्वं स्वीक्रियते। यदि कस्यचित् पदस्य असाधुत्वं प्राप्यते चेत् तद् समस्तपदं सम्यक् न विच्छेदितम् इति विचिन्त्य तस्य तत्रैव त्यागः करणीयः इति । 2) सर्वेषाम् समासविच्छेदितरूपाणाम् यद् अन्तिमं पदं स्यात् तेनैव समासविभक्त्यादिविचारः इति। उदाहरणार्थं अधः प्रदत्तं चित्रं दृश्यताम्।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.



समासपदविच्छेदप्रक्रियायाः सोपानानि

1. सन्धिनियमान् उपयुज्य प्रदत्तसमस्तपदस्य सर्वं सम्भावितं विच्छेदनं प्राप्तव्यम्।
2. पदविश्लेषकः इत्युपकरणम् उपयुज्य समस्तपूर्वपद-उत्तरपदयोः साधुत्वं प्राप्तव्यम्।
3. पूर्वस्मिन् प्रघट्टे यथा वर्णितं तथा प्रति समस्तपदस्य व्यवहारदृष्ट्या श्रेणिप्रदानम् ।
4. श्रेणिप्रदानानन्तरम् सर्वाणि समस्तरूपाणि क्रमशः स्थापनीयानि। तत्र प्रथमश्रेण्यां यत् समस्तं तस्यैव प्राधान्यम् इति।

परिणामः

पदविश्लेषकः[11] इत्युपकरणं चतुर्दशकोटिशब्दानां विश्लेषणं कर्तुं समर्थः। आहत्य 2,650 सन्धिनियमाः सङ्गृहीताः। तद्परीक्षणार्थं 8,260[12] उदाहरणानि समानन्तररूपेण स्वीकृतानि। परीक्षणानन्तरम् अस्माभिः यः परिणामः प्राप्तः। सः अधः प्रस्तूयते।

-प्रथमश्रेण्यां यानि समस्तपदविच्छेदितरूपाणि, तानि 92.5% समीचीनानि सन्ति। श्रेण्याम् आरम्भिकत्रिषु आगतानि यानि रूपाणि तानि 91.1% समीचीनानि सन्ति।

-सामान्यतया 15 अक्षरात्मकशब्दानां विच्छेदनार्थं यन्त्रं 0.04 क्षणात्मककालं स्वीकरोति।[13]

अधः प्रदत्ततालिकायां श्रेणिप्रदानविवरणमित्यादि दर्श्यते।

---

11 http://sanskrit.uohyd.ernet.in/~anusaaraka/sanskrit/samsaadhanii/morph/index.html

12 परिक्षणार्थम् उदाहरणानि महाभारतम् इति ग्रन्थात् स्वीकृतानि।

13 अस्माभिः 2.93GHz Core 2 Duo processor and 2GB RAM इति यन्त्रे परिक्षितम्।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

श्रेणी	%
1	92.46 35
2	5.05 92
3	1.62 35
4	0.29 78
5	0.19 36
>5	0.37 23

श्रेण्यानुसारेण प्रतिशद्दर्शनम्

5. समासपदान्वयः

समासपदच्छेदः पदविश्लेषकस्य सन्धिच्छेदकस्य च साहाय्येन एकस्य समस्तपदस्य एकं वा एकाधिकं वा छिन्नरूपं जनयति। तद्छिन्नरूपं स्वीकृत्य समासपदान्वयः द्वयोर्द्वयोः पदयोः समूहीकरणं करोति। एतत्समूहीकरणेन ज्ञायते यत् कियद्पर्यन्तं छिन्नरूपाणां समूहीकरणं कर्तुं शक्यत इति। उदाहरणार्थं त्रिपदात्मक-छिन्न-समस्तपदं वर्तते यथा पद 1-पद 2-पद 3 इति । सामान्यसमस्तपदे पदद्वयोरेव समावेशः दृश्यते । तथा चेत् पद 1-पद 2-पद 3 इत्यत्र द्विविधं समूहविभजनं कर्तुं शक्यते। यथा - <पद 1<पद 2-पद 3>> अथवा <<पद 1-पद 3> पद 3> इति। प्रसङ्गदृष्ट्या तु तयोः एक एव समीचिनः समूहः । स्पष्टार्थं प्रदत्ते उदाहरणे दृश्येताम्।

1) <बद्ध-<जाम्बूनद-स्रजः>>

2) <<तपस् -स्वाध्याय>-निरतम् >

त्रिपदात्मकसमस्तपदे तु केवलं द्विविधं समूहीकरणं सम्भवति इति दृष्टम्। परन्तु समस्तपदे यदा पदानां संख्या वर्धते तदा तु समूहानां संख्यापि एधिष्यते। यथा - यदि चतुष्पदात्मकः समासः स्यात् तर्हि तस्य पञ्चप्रकारेण समूहीकरणं सम्भवति। यदि पञ्चपदात्मकः समासः तर्हि चतुर्दशप्रकारेण समूहीकरणं सम्भवति। यदि षष्ठपदात्मकः समासः तर्हि तस्य द्वाचत्वारिंशत्प्रकारेण तत्समूहीकरणं सम्भवति इति। गणितशास्त्रे एतदर्थं साधारणरूपेण उच्यते यदि n+1 पदानि स्युः तदा तु 'केटालन् नम्बर' (Catalan number) इति सिद्धान्तम् उपयुज्य समूहीकरणं कर्तुं शक्यते इति। सः सिद्धान्तः एवं वर्तते -

$$C_n = (2n)!/(n+1)!n!$$

इति। सन्दर्भदृष्ट्या समीचीनसमूहस्य चयनमेव समासपदान्वयस्य कार्यम् इति । वस्तुतः एतादृशं कार्यं यान्त्रिकदृष्ट्या अतिमहत्त्वपूर्णं वर्तते। एतस्यां दिशि अपि कार्यं करणीयम्।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

6. समासभेदनिर्धारणम्

समासपदान्वयः एकवारं लभ्यते चेत् तदनन्तरम् एकैकं समूहं स्वीकृत्य तत्र कीदृशः समासः भवितुमर्हति इति निर्धारणमेव 'समासभेदनिर्धारकः' इत्युपकरणस्य कार्यम् इति। पूर्वोक्तरीत्या अस्माभि 55 समासानाम् द्वितीयातत्पुरुषेत्यादिरूपेण भेदाः सङ्गृहीताः। तत्र प्रतिभेदाय अनुबन्धनाम अपि प्रदत्तम् यथा T6 (षष्ठीतत्पुरुषः), Bs2 (द्वितीयाबहुव्रीहिः) इति।

एतत्कार्यसम्पादनाय अस्माभिः मानवीयानुबन्धितोदाहरणानि स्वीकृतानि। तत्र आहत्य 150 सहस्रपदानि वर्तन्ते। तेषु 12,796 समस्तपदानि[14] विद्यन्ते। एतानि उदाहरणानि परिशिष्टः-1 मध्ये प्रदत्तानुबन्धसूचिकानुसारेण अनुबन्धितानि। तत्र विद्यमानानि उदाहरणानि सन्दर्भदृष्ट्या अनुबन्धितानि। एतानि उदाहरणानि यन्त्रस्य प्रशिक्षणार्थं स्वीकृतानि। अन्यानि च 400 उदाहरणानि परीक्षणार्थं स्वीकृतानि।

मानवीयानुबन्धितोदाहरणानां विषये केचन अंशाः -

1) 10% उदाहरणानां पुनरावृत्तिः अपि वर्तते।

2) 12,976 समस्तपदेषु 2,630 विभिन्नानि पूर्वपदानि विद्यन्ते तथा च 7,106 विभिन्नानि उत्तरपदानि विद्यन्ते।

3) तालिका-2 मध्ये समासस्य अन्यभेदानां आवृत्तिसंख्याविषयकसूचनाः विद्यन्ते। तालिका-3 मध्ये आर्थिकदृष्ट्या समासभेदानां आवृत्तिसंख्याविषयकसूचनाः विद्यन्त इति।

अस्माभिः आवृत्तिसंख्यां प्राप्तुं अधः प्रदत्तसांख्यिकिनियमः प्रयुज्यते। तद्यथा -

P(T/पद 1-पद 2) = पद 1-पद 2 इति समासस्य, 'T' इति समासभेदः सम्भवितुम् अर्हति इति।

समासभेदः	शब्दाः (%)
T6	28.35
Bs6	12.45
K1	9.63
Tn	8.56
Di	7.23
U	5.73

तालिका-2

समासभेदः	शब्दाः (%)
T	52.43
B	18.96
K	12.04
D	8.84
U	5.73

तालिका-3

तर्हि

14 सामान्यसमस्तानि उदाहरणानि। यत्र द्वयोः द्वयोः पदयोः एव समासः।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

P (T/पद 1-पद 2) = P(T/पद 1)*P(T/पद 2)

यत्र

$$P(T/पद 1) = P(T*पद_i)/P(पद_i), i = 1.2$$

यत्र तु उपरितननियमैः किमपि न प्राप्यते तत्र अधः प्रदत्तनियमः प्रयुज्यते।

$$P(T.पद i) = 1/F,$$

$P(पद_i) = 1/F$ अत्र F इत्यक्तौ समासमधिकृत्य मानवीयानुबन्धितकृतशब्दानां सङ्ख्या।

6.1 मूल्याङ्कनम्

मूल्याङ्कनार्थम् अस्माभिः 400 समासोदाहरणानि स्वीकृतानि। तत्रापि यत्र द्वयोः पदयोरेव समासः तादृशोदाहरणानि एव स्वीकृत्य सन्दर्भं दृष्ट्वा अनुबन्धितानि। समासनामनिर्धारक इत्युपकरणन्तु सन्दर्भं न जानाति। तदर्थं तद् एकाधिकम् उत्तरं प्रदर्शयति। किम् उत्तरम् आदौ, किमनन्तरं च भवितव्यम् इति तादृशं क्रममपि निर्माति। सामान्यतया मूल्याङ्कने प्रदत्तोत्तराणां शुद्धता सीमा च द्रष्टव्या। परन्तु समासनामनिर्धारकस्तु प्रदत्तसमासपदस्य आवृत्तिसंख्यां दृष्ट्वा समासभेदं निर्धारयति। अत्र प्रदत्तक्रमः मुख्यः भवति। यतोहि क्रमं दृष्ट्वा एवास्माभिः तस्य शुद्धता इत्यादिनां विषये चिन्तयितुं शक्यते। अधः सामान्यरूपेण अस्माभिः मूल्याङ्कनं तालिकायां प्रदर्श्यते।

	55 समासभेदानां मूल्याङ्कनम्		8 समासभेदानां मूल्याङ्कनम्	
क्रमः	शब्दानां सङ्ख्या	शब्दाः (%)	शब्दानां सङ्ख्या	शब्दाः (%)
1	252	63	291	72.7
2	44	10.9	53	13.2
3	29	7.2	38	9.5

तालिका-4 समासनामनिर्धारकस्य मूल्याङ्कनसूचनाः

यदि वयं प्रथमक्रमस्य विद्यमानमूल्याङ्कनमेव स्वीकुर्मः चेत् तदा समासभेदनिर्धारकस्य 55 समासभेदानां शुद्धता 63% वर्तते, यदि केवलं 8 समासभेदानां स्वीकुर्मश्चेत् तदा समासभेदनिर्धारकस्य शुद्धता 72.7%. परन्तु यदि क्रमत्रयमपि स्वीकुर्मश्चेत तदानीं तस्य शुद्धता 95% भवति।

एतस्य कार्यदक्षतां सन्दर्भनियमान् संयोज्य इतौपि एधयितुं शक्यते। अष्टाध्यायां सामान्यतया 200 समासविधायकसूत्राणि विद्यन्ते । यथा - अन्नेन व्यञ्जनम् [2.1.34] इत्यनेन सूत्रेण व्यञ्जनवाचितृतीयान्तसुबन्तस्य अन्नवाचिसुबन्तेन सह विकल्पेन तत्पुरुषसमासः विधीयते । संस्कृत-समास-प्रक्रिया इत्यस्य अन्तर्जालसंस्करणमपि विद्यते। तद्द्रष्टुं कृपया अधः प्रदत्तसङ्केतः प्रयुज्यताम्।

http://sanskrit.uohyd.ernet.in/samAsa/frame.html

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

References

1. Bharati, A., Kulkarni, A.P., and Sheeba, V.: *Building a wide coverage Sanskrit Morphological Analyser: A practical approach.* In: The First National Symposium on Modelling and Shallow Parsing of Indian Languages, IIT-Bombay. (2006)

2. Bhat, G.M.: *Samāsaḥ* . Samskrita Bharati, Bangalore, Karnataka, 2006.

3. Fortes, F.C.L., and Roxas, R.E.O.: *Optimality Theory in Morphological Analysis* In: National Natural Language Processing Research Symposium, January 2004.

4. Fosler, J.E.: *On Reversing the Generation Process in Optimality Theory.* In : Proceedings of the Association for Computational Linguistics. (1996) .

5. Gillon, B.S.: *Exocentric Compounds in Classical Sanskrit.* In: Proceeding of the First International Symposium on Sanskrit Computational Linguistics(SCLS-2007), Paris, France, 2007.

6. Gillon, B.S.: *Tagging Classical Sanskrit Compounds.* In: Sanskrit Computational Linguistics 3, pages 98-105, Springer-Verlag LNAI 5406. (2009)

7. Hellwig, O.: *Sanskrit Tagger: A Stochastic Lexical and POS Tagger for Sanskrit.* In: Sanskrit Computational Linguistics 1 & 2, pages 266-277, Springer-Verlag LNAI 5402. (2007)

8. Hellwig, O.: *Extracting Dependency Trees from Sanskrit Texts.* In: Sanskrit Computational Linguistics 3, pages 106-115, Springer Verlag LNAI 5406. (2009)

9. Hoeks, J.C.J., and Hendriks, P.: *Optimality Theory and Human Sentence Processing: The Case of Coordination.* In: Proceedings of the 27th Annual Meeting of the Cognitive Science Society, Erlbaum, Mahwah, NJ, pp. 959–964. (2005)

10. Huet, G.: *Shallow syntax analysis in Sanskrit guided by semantic nets constraints.* In: Proceedings of International Workshop on Research Issues in Digital Libraries, Kolkata. (2006)

11. Huet, G.: Lexicon-directed Segmentation and Tagging of Sanskrit. XIIth World Sanskrit Conference, Helsinki, Finland, Aug. 2003. In Themes and Tasks in Old and Middle Indo-Aryan Linguistics, Eds. Bertil Tikkanen and Heinrich Hettrich. Motilal Banarsidass, Delhi, 2006, pp. 307-325.

12. Huet, G.: *Formal structure of Sanskrit text: Requirements analysis for a Mechanical Sanskrit Processor.* In: Sanskrit Computational Linguistics 1 & 2, pages 162-199, Springer-Verlag LNAI 5402. (2007)

13. Huet, G.: *Sanskrit Segmentation.* In: South Asian Languages Analysis Roundtable XXVIII, Denton, Ohio. (October 2009)

14. Jha, G.N. and Mishra, S.K.: *Semantic Processing in Pāṇini's kāraka System.* In: Sanskrit Computational Linguistics 1 & 2, pages 239-252, Springer-Verlag LNAI 5402. (2007)

15. Kulkarni, A.P., Shukla, D.: *Sanskrit Morphological Analyser*: Some Issues. In: To appear in Bh.K

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

Festschrift volume by LSI. (2009)

16. Kulkarni, A.P., Kumar, A., Sheeba, V.: *Sanskrit compound paraphrase generator.* In: Proceedings of ICON-2009: 7th International Conference on Natural Language Processing, Macmillan Publishers, India.

17. Mahavira: *Pāṇini as Grammarian* (With special reference to compound formation). Bharatiya Vidya Prakashan [Delhi - Varanasi], India, June 1978.

18. Mittal, V.: *Automatic Sanskrit Segmentizer using Finite State Transducers.* In: Proceeding of Association for Computational Linguistics - Student Research Workshop. (2010)

19. Murty, M.S.: *Sanskrit Compounds-A Philosophical Study.* Chowkhamba Sanskrit Series Office, Varanasi(India), 1974.

20. Pandit Ishvarachandra: Aṣṭādhyāyī Chaukhamba Sanskrit Pratisthan, Delhi, 2004.

21. Prince, A., Smolensky, P.: *Optimality Theory: Constraint Interaction in Generative Grammar.* In: RuCCS Technical Report 2 at Center for Cognitive Science, Rutgers University, Piscataway. (1993)

22. Sarma, E.R.S.: *Maṇikaṇa : A Navya-Nyāya Manual.* The Adyar Library and research Centre, Madras. (1960)

23. Scharf, P.M.: *Levels in Pāṇini's Aṣṭādhyāyī* In: Sanskrit Computational Linguistics 3, pages 66-77, Springer-Verlag LNAI 5406. (2009)

24. Yuret, D., Bicici, E.: *Modeling Morphologically Rich Languages Using Split Words and Unstructured Dependencies.* In: ACL-IJCNLP, Singapore. (2009)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

परिशिष्टः-अ : संस्कृतसमासभेदाः

अव्ययीभावः		कर्मधारयः	
समासभेदाः	अनुबन्धनाम	समासभेदाः	अनुबन्धनाम
अव्यय-पूर्वपदः	A1	विशेषण-पूर्वपद-कर्मधारयः	K1
अव्यय-उत्तरपदः	A2	विशेषण-उत्तरपद-कर्मधारगः	K2
तिष्ठद्गुप्रभृतिः	A3	विशेषण-उभयपद-कर्मधारयः	K3
संख्यापूर्वद-नद्युत्तरपदः	A4	उपमान-पूर्वपद-कर्मधारयः	K4
नद्युत्तरपद-अन्यपदार्थसंज्ञायाम्	A5	उपमान-उत्तरपद-कर्मधारयः	K5
संख्यापूर्वपद-वंश्योत्तरपदः	A6	अवधारणापूर्वपद-कर्मधारयः	K6
पारे-मध्ये-पूर्तपदषष्ठ्युत्तरपदः	A7	सम्भावनापूर्वपद-कर्मधारयः	K7
		मध्यमपदलोपिकर्मधारयः	Km

तत्पुरुषः		बहुव्रीहिः	
समासभेदाः	अनुबन्धनाम	समासभेदाः	अनुबन्धनाम
प्रथमातत्पुरुषः	T1	द्वितीयार्थबहुव्रीहिः (समानाधिकरणः)	Bs2
द्वितीयातत्पुरुषः	T2	तृतीयार्थबहुव्रीहिः (समानाधिकरणः)	Bs3
तृतीयातत्पुरुषः	T3	चतुर्थ्यर्थबहुव्रीहिः (समानाधिकरणः)	Bs4
चतुर्थीतत्पुरुषः	T4	पञ्चम्यर्थबहुव्रीहिः (समानाधिकरणः)	Bs5
पञ्चमीतत्पुरुषः	T5	षष्ठ्यर्थबहुव्रीहिः (समानाधिकरणः)	Bs6
षष्ठीतत्पुरुषः	T6	सप्तम्यर्थबहुव्रीहिः (समानाधिकरणः)	Bs7
सप्तमीतत्पुरुषः	T7	दिग्वाचक-बहुव्रीहिः (समानाधिकरणः)	Bsd
नञ्तत्पुरुषः	Tn	प्रहरणविषयक-बहुव्रीहिः (समानाधिकरणः)	Bsp
समाहारद्विगुः	Tds	ग्रहणविषयक-बहुव्रीहिः (समानाधिकरणः)	Bsg
तद्धितार्थद्विगुः	Tdt	अस्त्यर्थ-मध्यमपदलोपि(नञ्)-बहुव्रीहिः	Bsmn
उत्तरपदद्विगुः	Tdu	प्रादि-बहुव्रीहिः	Bvp
गतिसमासः	Tg	संख्योभयपद-बहुव्रीहिः (समानाधिकरणः)	Bss

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

कुसमासः	Tk	उपमानपूर्वपद-बहुव्रीहिः (समानाधिकरणः)	Bsu
प्रादिसमासः	Tp	व्यधिकरण-बहुव्रीहिः	Bv
मयूरव्यंस्कादिसमासः	Tm	सङ्ख्योत्तरपद-व्यधिकरण-बहुव्रीहिः	Bvs
तत्पुरुषः बहुपदः	Tb	सहपूर्वपद-व्यधिकरण-बहुव्रीहिः	BvS
तत्पुरुषः उपपदः	U	उपमानपूर्वपद-व्यधिकरण-बहुव्रीहिः	BvU
		बहुपद-बहुव्रीहिः	Bb

द्वन्द्वः		केवल	
समासभेदाः	**अनुबन्धनाम**	**समासभेदाः**	**अनुबन्धनाम**
इतरेतरयोग-द्वन्द्वः	Di	केवलसमासः	S
समाहार-द्वन्द्वः	Ds	**द्विरुक्तिः**	
एकशेषः	E	द्विरुक्तिः	d

परिशिष्टः - ब : समासविग्रह-विधायक-नियमाः

	अव्ययीभावः
1	<x-y>A1 => y{6} f{x} where f maps x to the noun with same semantic content. A function f needs to be defined.
2	<x-y>A2 => x{3} विपरीतं वृत्तम्
3	<x-y>A4 => x{6} y{6} समाहारः
4	<x-y>A5 => x'{1} y{1} यस्मिन् देशे x' y' इत्यनयोः समानलिङ्गं स्यात् इति।
5	<x-y>A6 => x{6} y{6} समाहारः यदि x=द्वि स्यात् तर्हि x एवं y इत्यनयोः प्रयोगः द्विवचने स्यात्।
6	<x-y>A7 => x{6} y
	तत्पुरुषः
7	<x-y>T_n => x{n} y $2 \le n \le 7$
8	<x-y>Tn => x{n} y

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

9	<x-y>Tds => x{6;ba} y{6;ba} समाहारः
10	<x-y-z>Tb => x{1} y{1} z{1}
	कर्मधारयः
11	<x-y>K1 => x{1} तत् y{1} च
12	<x-y>K2 => x{1} च y{1} च
13	<x-y>K3 => x{1} च असौ y{1} च
14	<x-y>K4 => x{1} इव y{1}
15	<x-y>K5 => x{1} y{1} इव
16	<x-y>K6 => x{1} एव y{1}
17	<x-y>K7 => x{1} इति y{1}
	बहुव्रीहिः
18	<x-y>Bsd => x{6} च y{6} च यदन्तरालम्
19	<x-y>Bsp => x{3} च y{3} च प्रहृत्य इदं युद्धं प्रवृत्तम्
20	<x-y>Bsg => x{7} y{7} गृहीत्वा इदं युद्धं प्रवृत्तम्
21	<x-y>Bsmn => x' y{1} यस्य
22	<x-y>Bss => x{1} वा y{1} यस्य
23	<x-y>Bsu => x{1} इव y{1} यस्याः
24	<x-y>Bv => x y{1} यस्य
25	<x-y>Bvs => y{6} x' ये सन्ति ते
26	<x-y>BvS => y{3} सह
27	<x-y>BvU => x{6} इव y यस्य
	द्वन्द्वः
28	<x-y>Di => x{1} च y{1} च
29	<x-y>Ds => x{1} च y{1} च
30	<x-y>S => y{1} x{1}
31	<x-y>d => x y

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri

ICON 2009 Dec

(125)

Sanskrit Compound Paraphrase Generator

Anil Kumar[1], V.Sheebasudheer[2], Amba Kulkarni[3]

[1]Department of Sanskrit Studies, University of Hyderabad, Hyderabad,
anil.lalit22@gmail.com,
[2]Chinmaya International Foundation Shodhasansthan, Veliyanad, Kerala,
v.sheeba@gmail.com,
[3]Dept. of Sanskrit Studies, University of Hyderbad, Hyderabad,
ambapradeep@gmail.com

Abstract

Sanskrit is very rich in compound formation unlike modern Indian Languages. The compound formation being productive it forms an open-set and as such it is also not possible to list all the compounds in a dictionary. The compound formation involves a mandatory sandhi. But mere sandhi splitting does not help a reader in identifying the meaning of a compound, since typically a compound does not code the relation between its components explicitly. To understand the meaning of a compound, it is necessary to identify its components and discover the relation between them. An expression providing the meaning of a compound is called a paraphrase.

In this paper, we summarize our experience in building an automatic paraphrase generator for Sanskrit compounds. On the face of it, it simply turns out to be a generation problem and hence should be trivial. The actual implementation, however needed a morphological analyser that can handle the bound morphemes in compounds. The paraphrase handler could produce correct paraphrases for around 90% of the compounds.

1 Introduction

Sanskrit has more than 2500 years old almost exhaustive grammar in the form of Pāṇini's Aṣṭādhyāyī which has the features of computationality. However, only recently Sanskrit Computional Linguistics[1] has gained a momentum. In this decade there have been many efforts to develop computational tools to access Sanskrit texts ([5], [7], [12], [8], [4]). There have also been ongoing efforts to compare the perforumances of various morph analysers and also to discuss the interoperability issues, both at the national as well as international levels.

Most of the Sanskrit Computational tools handle morphological analysis and sandhi splitting. Some of them also do the sentential parsing. Sanskrit is very rich in compound formation unlike modern Indian Languages. The compound formation being productive it forms an open-set and as such it is also not possible to list all the compounds in a dictionary. The compound formation involves a mandatory sandhi. But mere sandhi splitting does not help a reader in identifying the meaning of a compound, since typically a compound does not code the relation between its components explicitly. To understand the meaning of a compound, it is necessary to identify its components and discover the relation between them. An expression providing the meaning of a compound is called a paraphrase (or विग्रहवाक्य).

In this paper we summarize our experience in building an automatic paraphrase generator for Sanskrit compounds. The organisation of the paper is as follows: We give a brief outline of salient features of Sanskrit compounds and its semantic classification in the next section. In the third section, we describe the actual algorithms. The results of evaluation are presented in the fourth section.

[1]International Symposium on Sanskrit Computational Linguistics held in 2007, 2008 and 2009.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

(126)

2 Sanskrit Compounds

Typical Western Linguistics definition of compound is a "lexeme with more than one stems". The Sanskrit word समासः(samāsaḥ) for a compound means समसनम्(samasanam) which means "combination of more than one words into one word which conveys the same meaning as that of the collection of the component words together". While combining the components together, a compound undergoes certain operations such as loss of case suffixes, loss of accent etc. A Sanskrit compound thus has one or more of the following features ([9], [3]):

1. It is a single word (एकपदम्).
2. It has a single case suffix (एकविभक्तिकम्) with an exception of अलुक्(aluk) compounds such as युधिष्ठिरः(yudhiṣṭiraḥ), where there is no deletion of case suffix of the first component.
3. It has a single accent(एकस्वरः).
4. The order of components in a compound is fixed.
5. No words can be inserted in between the compounds.
6. The compound formation is binary with an exception of dvandva and bahupada bahuvrīhi.
7. Euphonic change (Sandhi) is a must in a compound formation.
8. Constituents of a compound may require special gender or number different from thier default gender and number. e.g. pāṇipādam, pācikābhāryaḥ, etc.

Though compounds of 2 or 3 words are more frequent, Sanskrit literature is rich in compounds involving more than 3 constituent words with some compounds even running through pages. Here are some examples of Sanskrit compounds involving more than 3 words.

1. वेदवेदाङ्गतत्त्वज्ञः[2]=वेद-वेदाङ्ग-तत्त्व-ज्ञः(a person knowledgable in the philosophies/essence of the veda and vedāṅga)

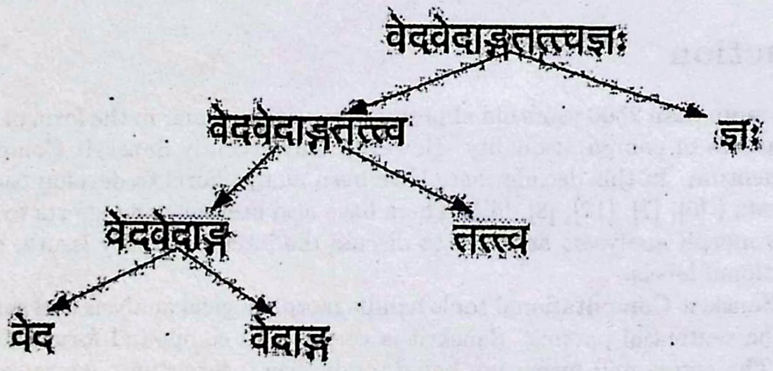


Figure 1: Binary representation of वेदवेदाङ्गतत्त्वज्ञः

2. प्रवरमुकुटमणिमरीचिमञ्जरीचयचर्चितचरणयुगल[3]=प्रवर-मुकुट-मणि-मरीचि-मञ्जरी-चय-चर्चित-चरण-युगल
3. जलादिव्यापकपृथिवीत्वाभावप्रतियोगिपृथिवीत्ववती[4]=जल-आदि-व्यापक-पृथिवीत्व-अभाव-प्रतियोगि-पृथिवीत्ववती

The Sanskrit compounds are classified semantically into four major types:

- तत्पुरुषः(Endocentric with head typically to the right),

[2]Rāmāyaṇam 1-1-14
[3]Pañcatantram in kathāmukham
[4]Kevalavyatireki-prakaraṇam : Maṇikaṇa

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri

- बहुव्रीहि: (Exocentric),
- द्वन्द्व: (Copulative), and
- अव्ययीभाव: (Endocentric with head typically to the left and behaves as an indeclinable).

The compounds are formed with two words at a time with an exception of द्वन्द्व(dvandva)[5] and बहुपद-बहुव्रीहि (bahupada bahuvrīhi). Hence compounds can be represented faithfully as a binary tree, as shown in figure 1.

3 Compound Processor

There are three tasks involved in identifying the meaning of a compound.

1. **Syntactic Parser:** To get the parse of a compound depicting its composition, in terms of a tree. A Syntactic parser will produce the parse of an input string तपस्स्वाध्यायनिरतेन as <<तपस्-स्वाध्याय>-निरतेन>, or equivalantly as shown in figure 2.

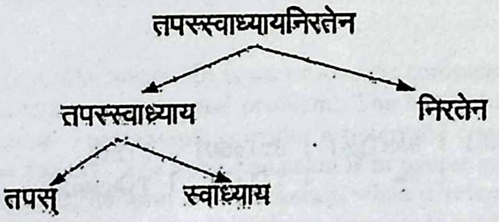


Figure 2: Syntactic Parser

2. **Semantic analyser:** To decide the type of a compound at each node of composition. A Semantic analyser would assign labels to each of these nodes as <<तपस्-स्वाध्याय>Di-निरतेन>T7 or as shown in figure 3,

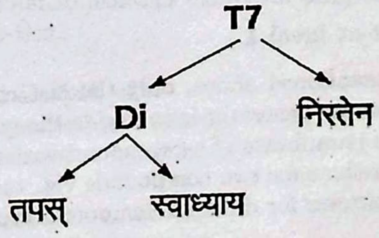


Figure 3: Semantic analyser

where **T7** and **Di** for the compound labels viz. सप्तमी-तत्पुरुष(saptamī-tatpuruṣa) and इतरेतर-द्वन्द्व(itaretara dvandva).

3. **Paraphrase Generator:** To generate a paraphrase of a compound. The Paraphrase generator would paraphrase the given compound as
तप: च स्वाध्याय: च = तपस्स्वाध्याय: (= तत्1)[6]

[5] An example of dvandva compound involving more than two constituents is रामलक्ष्मणभरतेभ्य: (rāmalakṣmaṇabharatebhyaḥ) which means for or from rāma, lakṣmaṇa and bharata.

[6] The pronoun तत् is used to indicate a variable, the indices 1,2,.. indicate different intances of the variable.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

gloss: penance and self-study
(तत्1{7}[7]=) तस्मिन् निरतः = तपस्स्वाध्यायनिरतः(=तत्2)
gloss: who is constantly engaged in penance and self-study
(तत्2{3}=) तेन = तपस्स्वाध्यायनिरतेन
gloss: By that person who is constantly engaged in penance and self-study

The work on syntactic parser involves sandhi splitter and this work is still in progress and not reported in this paper. The semantic analyser requires a lexical resource rich with semantic information. Pāṇini has given a set of rules which help in deciphering the compounds. These rules are being explored further. A tagging scheme to assign appropriate tags to the nodes is evolved. Two important aspects for deciphering a compound are to know the head and to know the relation between components. The proposed tags mark these two. E.g. **T7** stands for compound of type tatpuruṣa (तत्पुरुषः) and 7 indicates the vibhakti of the first component. Assuming that a manually parsed compound is available, we describe below a paraphrase generator.

3.1 Paraphrase generator

A semantically analysed compound has the following syntax.

```
compound: < component - component > tag
    ;
component: word
    | compound
    ;
tag: A[1-6]
    | Bs[2-7]  | Bs[dgpsu]  | Bsm[gn]  | Bv[sSU]  | B[bv]
    | D[is]    | K[1-5]     | Km       | T[1-7]   | T[bgmnp]
    | Tds      | [ESUd]
    ;
word; [a-zA-Z]+
Sanskrit compounds tagging sytax
```

We define level 1 of a tagged compound as the one which involves word and/or stem. The level 2 and above involve at least one component a compound. Thus a composition of components involving leaf nodes of a tree is a level 1 compound and a compound involving at least one non leaf node is a compound of level 2.

3.1.1 Paraphrase Generation at level 1

In Sanskrit compounds, as mentioned above, only the last component contains the case suffix[8]. Hence the major task of the paraphrase generator is to decide the gender, number and case suffix of the last component of compound. The paraphrase of a compound varies with the type of a compound. We illustrate below the rules and algorithms for two compounds viz. tatpuruṣa and bahuvrīhi. Appendix I contains rules for generating paraphrases for different compound types.

1. **Algorithm for tatpuruṣaḥ:** Input: <x-y>Tn
   where **x** is nominal stem (प्रातिपदिकम्),
   y is a noun,
   T stands for the tatpuruṣa compound and **n** indicates the case.
   Output: The paraphrase of <x-y>Tn
   Steps:

[7] {7} indicates 7^{th} case
[8] with an exception of an aluk samāsaḥ

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

(a) Get default gender of x from the lexicon. Let it be g_1.

(b) Get the morphological analysis for y. In case of multiple answers, use the following heuristics to select the most appropriate analysis.
-In case of different stems and different genders, keep all answers.
-In case of same stems with different genders, select the default gender.
Let the stem and the gender be y' and g_2 respectively. Let the case suffix be c_1.

(c) Generate the appropriate forms of individual components, by invoking the generator. Let
x= x<gen=g_1> <case=n> <num=1>,
and
y=y' <gen=g_2> <case=1> <num=1>

(d) Call the Sentence Generator to generater xy as the paraphrase of the compound <x-y>T_n.

(e) If y is not in nominative form, generate the singular form with case c_1 of the 3rd person pronoun in gender g_2.
For example the paraphrase of

<तपस्स्वाध्याय-निरतेन>T7

would be

तपस्स्वाध्याय{7}-निरतम्=तपस्स्वाध्याययोः निरतः, तेन तपस्स्वाध्यायनिरतेन

2. **Algorithm for bahuvrīhi :** The bahuvrīhi is an exocentric compound. As such the morphological analysis of such compounds pose a special problem. The head of such compounds assume the gender of an object it denotes. For example consider a bahuvrīhi compound formed by the words *pīta* (yellow) and *ambaram* (cloth). The word ambaram is in neuter gender when it is used to refer to a man wearing yellow clothes, its form is pītāmbaraḥ, when it refers to a woman wearing yellow clothes its form is pītāmbarā. The compounds in these cases are tagged as <pīta-ambaraḥ>Bs6 and <pīta-ambarā>Bs6. Now to generate the paraphrase, we require the nominal base (प्रातिपदिकम्) of *ambaraḥ* and *ambarā*. But since the dictionary does not have the base form with these genders, an analyser may fail to analyse the head. In order to handle such cases, the morphological analyser has been modified to get the default प्रातिपदिकम् in case the word is a post component (*uttarapada*) of a compound. We describe below the algorithm for generation of paraphrase of bahuvrīhi compounds.
Input <x-y>Bsn
where x is the nominal stem,
y is a noun,
Bs stands for the samānādhikaraṇa bahuvrīhi compound and **n** indicates the case.
Output: Paraphrase of <x-y>Bsn.
Steps:
a) Get the morphological analysis of y.
Let it be stem=y', gender=g_2, case=c_1.
Get the default gender of y. Let it be g_1.
b) Generate
x'=x <gen=g_1><num=1><case=1>
y"=y'<gen=g_2><num=1><case=1>
c) Let z be the 3^{rd} person pronoun with gender g_1.
Generate
z'=z<gen=g_1><num=1><case=c_1>
and
z"=z<gen=g_1><num=1><case=1>

The paraphrase is x' y" z' z"
e.g. <पीत-अम्बरः>Bs6=पीतम् अम्बरं यस्य सः,
<पीत-अम्बरा>Bs6=पीतम् अम्बरं यस्याः सा.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

3.1.2 Paraphrase Generation for higher level compounds

While generating a paraphrase of a compound involving another compound as a component, decision has to be made as to whether to use the compound or its paraphrase in the generation. To make the point clear consider a 2 level compound

तपस्स्वाध्यायनिरतेन=<तपस्स्वाध्याय-निरतेन>T
=<<तपस्-स्वाध्याय> *Di*–निरतेन>T7

The paraphrase of <तपस्-स्वाध्याय>Di is तपः च स्वाध्यायः च. The paraphrase of the complete compound then can be expressed as

a) तपः च स्वाध्यायः च=तपस्स्वाध्यायौ,
तपसि च स्वाध्याये च निरतः=तपस्स्वाध्यायनिरतः
तेन तपस्स्वाध्यायनिरतेन

or as

b) तपः च स्वाध्यायः च=तपस्स्वाध्यायौ,
तपस्स्वाध्याययोः निरतः=तपस्स्वाध्यायनिरतः
तेन तपस्स्वाध्यायनिरतेन

or as

c) तपः च स्वाध्यायः च=तपस्स्वाध्यायौ(=तत्1),
तस्मिन् निरतः=तपस्स्वाध्यायनिरतः(=तत्2),
तेन तपस्स्वाध्यायनिरतेन

From generation point of view **b** is easier than **a** and **c** is more easier then **b**. We have decided to follow **c**, not only becuase of its simplicity but because of its wide usage in Sanskrit literature as well. The only way we differ from the tradition is in indexing the pronouns if more than one components are present.

Following this approach, we give below the paraphrase of
प्रवर-मुकुट-मणि-मरीचि-मञ्जरी-चय-चर्चित-चरण-युगलः as
प्रवरराजानां मुकुटः = प्रवरमुकुटः (=तत्1),
(तत्1{6}=) तस्य मणिः = प्रवरमुकुटमणिः (=तत्2),
(तत्2{6}=) तस्य मरीचिः = प्रवरमुकुटमणिमरीचिः (=तत्3),
(तत्3{1}=) ते एव मञ्जर्यः = प्रवरमुकुटमणिमरीचिमञ्जर्यः (=तत्4),
(तत्4{6}=) तेषां चयः = प्रवरमुकुटमणिमञ्जरीचयः (=तत्5),
(तत्5{3}=) तेन चर्चितम् = प्रवरमुकुटमणिमञ्जरीचर्यचर्चितम् (=तत्6),
चरणस्य युगलम् = चरणयुगलम् (=तत्7),
(तत्6{1} तत्7{1}=) तत्6 तत्7 यस्य सः = प्रवरमुकुटमणिमरीचिमञ्जरीचयचर्चितचरणयुगलः

3.1.3 Problem Cases and their solutions

- **अलुक् समासः-**
  Only the last component of a compound typically has a case suffix. However as noted earlier there are exceptions. Such compounds are called aluk(अलुक्) samāsa. The current tagset of the Sanskrit Consortium does not mark aluk samāsas. In case of paraphrase generation, therefore it is necessary to check whether the pre-component is in the base form or not.

- **मध्यमपदलोपिः-**
  This is special type of compound in which some of the words in the paraphrase do not occur in its compound form. e.g. देवब्राह्मणः(Devabrāhmaṇaḥ) is a compound whose paraphrase is देवपूजकः ब्राह्मणः. So to get the paraphrase of such compounds mere components are not sufficient. One should also know the context to supply the missing words. We do not handle such compounds in the present implementation. These compounds being very rare not handling them does not affect the overall coverage of the paraphrase generator much.

- **Special cases from Gaṇapāṭa etc.:-**
  Compounds with special paraphrases have been listed by Pāṇini separately in a list. Examples of such compounds are मयूरव्यंसकः(Mayuravaymsakaḥ), कम्बोजमुण्डः(Kambojamuṇḍaḥ), यवनमुण्डः

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri

(Yavanamuṇḍaḥ) etc. Each one of them have a special paraphrase. Readymade paraphrases of such compounds are provided.

- उपपद-समासः-

  An upapada tatpuruṣa samāsaḥ has a verbal noun (kṛdanta) as a post component(e.g. ज्ञः and कारः in तत्त्वज्ञः and कुम्भकारः respectively). These forms are special and occur only as bound forms in a compound. Hence, a special morphological analyser to handle these forms is necessary. The current implementation does not handle these forms as well.

4 Evaluation

We have tested 160 examples from Saṅkṣepa Rāmāyaṇam. The performance of tested examples is given below.

	No. of examples	%
Correct Paraphrase	145	90.60
Wrong/Incomplete Paraphrase	15	9.40

The wrong or incomplete paraphrase is mainly because of the dictionary incompleteness. A morph analyser that guesses the stem can improve the system further. Another challenge we faced in the generation of paraphrase is related to gender.

Bahuvrihī compounds which are exocentric, typically are adjectives. Hence their paraphrases will change as per the gender of the nouns they qualify. The compound tagging naturally does not code this information, and hence the paraphrase (from the Sanskrit grammar point of view) may be defective. e.g. consider the word नीलवर्णम् as used in

1. इदं नीलवर्णं फलम्
   (This is a blue coloured fruit.)
2. नीलवर्णं फलम् आनय
   (Bring a blue coloured fruit.)
3. नीलवर्णं कृष्णं पश्य
   (See the blue coloured Krishna.)

In all these examples नीलवर्णम् is a ṣaṣṭī bahuvrīhi (Bs6) but is in nominative case in (1), and accusative case in (2) and (3), further it is neuter in (1) and (2) and masculine in (3). Hence the paraphrase will be different in all the three cases, viz.

1. नीलं वर्णं यस्य तत्, तत्[9]
2. नीलं वर्णं यस्य तत्, तत्[10]
3. नीलं वर्णं यस्य सः, तम्

Thus to generate a correct paraphrase, we need the context and sentence level analysis viz. the adj-noun relation (to get gender of विशेषणम्) and also the Kāraka role of the noun (to know the विभक्ति) in case the word form is ambiguous.

[9] Nominative case
[10] Accusative case

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

5 Future Work

The next task is to develop a syntactic parser and semantic analyser. A syntactic parser requires a sandhi splitter and the semantic analyser needs a lexicon rich with semantic features. It will be very interesting to look at the Pāṇini's rules that govern the semantic aspects of the compounds. There are around 200 rules in the Aṣṭādhyāyī dealing with the semantic aspect of the Sanskrit compounds. It may provide some insight into various kinds of features the lexicon should be enriched with in order to be able to carry out the semantic analysis mechanically.

Acknowlegement

We would like to thank K.V. Ramakrishnamacharyulu, K. Narayana Murty, J.S.R.A Prasad, Devanand Shukl and Sheetal Pokar for their valuable guidance and suggestions to improve the system.

This work is being carried out as a part of consortium project entitled "Development of Computational Toolkit and Sanskrit-Hindi Machine Translation System" sponsored by MCIT(DIT).

References

[1] G. Mahabaleshvara Bhat, *Samāsaḥ*. Samskrita Bharati, Bangalore, Karnataka, 2006.

[2] Brendan S. Gillon , *Exocentric Compounds in Classical Sanskrit.* Proceeding of the First International Symposium on Sanskrit Compoutational Linguistics(SCLS-2007), Paris, France, 2007.

[3] Brendan S. Gillon, *Tagging Classical Sanskrit Compounds.* Proceeding of the Third International Symposium on Sanskrit Compoutational Linguistics(SCLS-2009), Hyderabad, India, 2009.

[4] Oliver Hellwig, *Sanskrit Tagger: A Stochastic Lexical and POS Tagger for Sanskrit.* Proceeding of the International Symposium on Sanskrit Compoutational Linguistics(SCLS-2007), Paris, France, 2007.

[5] Gérard Huet, *Formal Structure of Sanskrit Text: Requirement Analysis for a Mechanical Sanskrit Processor.* Proceeding of the International Symposium on Sanskrit Computational Linguistics(SCLS-2007), Paris, France, 2007.

[6] Pandit Ishvarachandra, *Aṣṭādhyāyī.* Chaukhamba Sanskrit Pratisthan, Delhi, 2004.

[7] Girish Nath Jha and Sudhir K. Mishra, *Semantic Processing in Pāṇni's Kāraka System.* Proceeding of the International Symposium on Sanskrit Computational Linguistics(SCLS-2007), Paris, France, 2007.

[8] Amba Kulkarni and V. Sheeba, *Building a wide coverage Morphological Analyser for Sanskrit: A practical approach.* Invited speech at 'First National Symposium on Modeling and Shallow Parsing of Indian Languages', 31st March - 4th April 2006, IIT Mumbai.(http://sanskrit.uohyd.ernet.in/faculty/amba/)

[9] Mahavira, *Pāṇini as Grammarian (With special reference to compound formation).* Bharatiya Vidya Prakashan [Delhi - Varanasi], India, June 1978.

[10] M. Srimannarayana Murty, *Sanskrit Compounds-A Philosophical Study.* Chowkhamba Sanskrit Series Office, Varanasai(India), 1974.

[11] Vasudeva Lakshman Shastri Paniskar, *Sidhānta Kaumudi.* Meharchand Lachamandas Publications, New-Delhi, 1985.

[12] Peter M. Scharf, *Levels in Pāṇini's Aṣṭādhyāyī.* Proceeding of the International Symposium on Sanskrit Compoutational Linguistics(SCLS-2009), University of Hyderabad, Hyderabad, India, 2009.

[13] V. Sheebasudheer, Amba Kulkarni, K.V.Ramakrishnamacharulu , *Guidelines for tagging of Sanskrit Compounds.* (Internal Circulation).

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

6 Appendix

	अव्ययीभावः
1	<x-y>A1 => y{6} f{x} where f maps x to the noun with same semantic content. A function f needs to be defined.
2	<x-y>A2 => x{3} विपरीतम् वृत्तम्
3	<x-y>A4 => x{6} y{6} समाहारः
4	<x-y>A5 => x'{1} y{1} यस्मिन् देशे x' has same gender as that of y'.
5	<x-y>A6 => x{6} y{6} समाहारः if x = द्वि, both x and y will be in द्विवचनम्
6	<x-y>A7 => x{6} y
	तत्पुरुषः
7	<x-y>T_n => x{n} y $2 \leq n \leq 7$
8	<x-y>Tn => x{n} y
9	<x-y>Tds => x{6;ba} y{6;ba} समाहारः
10	<x-y-z>Tb = x{1} y{1} z{1}
	कर्मधारयः
11	<x-y>K1 => x{1} तत् y{1} च
12	<x-y>K2 => x{1} च y{1} च
13	<x-y>K3 => x{1} च असौ y{1} च
14	<x-y>K4 => x{1} इव y{1}
15	<x-y>K5 => x{1} y{1} इव
16	<x-y>K6 => x{1} एव y{1}
17	<x-y>K7 => x{1} इति y{1}
	बहुव्रीहिः
18	<x-y>BS_n => x{1} y{1} यत्{g}{n} where g is the gender of y $2 \leq n \leq 7$
19	<x-y>Bsd => x{6} च y{6} च यदन्तरालम्
20	<x-y>Bsp => x{3} च y{3} च प्रहृत्य इदम् युद्धम् प्रवृत्तम्
21	<x-y>Bsg => x{7}-y{7}+ गृहीत्वा इदम् युद्धम् प्रवृत्तम्
22	<x-y>Bsmn => x'-y{1} यस्य
23	<x-y>Bss = > x{1} वा y{1} यस्य
24	<x-y>Bsu => x{1} इव y{1} यस्याः
25	<x-y>Bv => x y{1} यस्य
26	<x-y>Bvs => y{6} x' ये सन्ति ते
27	<x-y>BvS => y{3} सह
28	<x-y>BvU => x{6} इव y यस्य
	द्वन्द्वः
29	<x-y>Di => x{1} च y{1} च
30	<x-y>Ds => x{1} च y{1} च
31	<x-y>S => y{1} x{1}
32	<x-y>d => x y

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

Vibhakti Divergence between Sanskrit and Hindi

Preeti Shukla, Devanand Shukl and Amba Kulkarni

Department of Sanskrit Studies,
University of Hyderabad,
Hyderabad
shukla.preetidev@gmail.com,dev.shukl@gmail.com,apksh@uohyd.ernet.in

Abstract. Translation divergence at various levels between languages arises due to the different conventions followed by different languages for coding the information of grammatical relations. Though Sanskrit and Hindi belong to the same Indo-Aryan family and structurally as well as lexically Hindi inherits a lot from Sanskrit, yet divergences are observed at the level of function words such as vibhaktis. Pāṇini in his Aṣṭādhyāyī has assigned a default vibhakti to kārakas alongwith many scopes for exceptions. He handles these exceptions either by imposing a new kāraka role or by assigning a special vibhakti. However, these methods are not acceptable in Hindi *in toto*. Based on the nature of deviation, we propose seven cases of divergences in this paper.

Key words: Pāṇini, kāraka, vibhakti, semantics, translation divergence.

1 Introduction

Different languages may follow different conventions for coding the information of grammatical relations. This leads to divergence between languages at various levels in translation in general and in Machine Translation(MT) in particular. Translation divergence patterns have been discussed by Dorr [Dorr,1994], and based on those patterns considerable work is done for Indian Languages, viz., English-Sanskrit-Hindi MT [Goyal and Sinha, 2009], English-Sanskrit MT [Mishra and Mishra, 2008], English-Hindi MT [Dave *et al.*, 2002] to name a few.

Sanskrit and Hindi belong to the same Indo-Aryan family and Hindi inherits a lot from Sanskrit in terms of structure as well as lexicon. As such, when we look at the divergences between Sanskrit and Hindi, most of the cases discussed in Dorr's divergence turn out to be either of rare occurrence, or do not pose much problem as far as accessing the source text using machine tools is concerned. On the contrary divergences are observed at the level of function words such as vibhaktis. While both Sanskrit and Hindi are inflectional languages, Sanskrit is synthetic in nature while Hindi is analytic [Dwivedi, 2006]. In Sanskrit both *sup* and *tiṅ* suffixes are termed as vibhakti pratyayas.[1] *Sup* suffix is added to a nominal stem and *tiṅ* suffix is added to a verbal root. But in Hindi only the

[1] vibhaktiśca (suptiṅau vibhaktisanjñau staḥ -S.K.) (P-1.4.104)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri

(136)

suffix added to the nominal stem is called as a vibhakti pratyaya. In this paper we discuss the divergences in the use of nominal suffixes (vibhakti pratyayas) between Sanskrit and Hindi.

2 Classification of Divergences

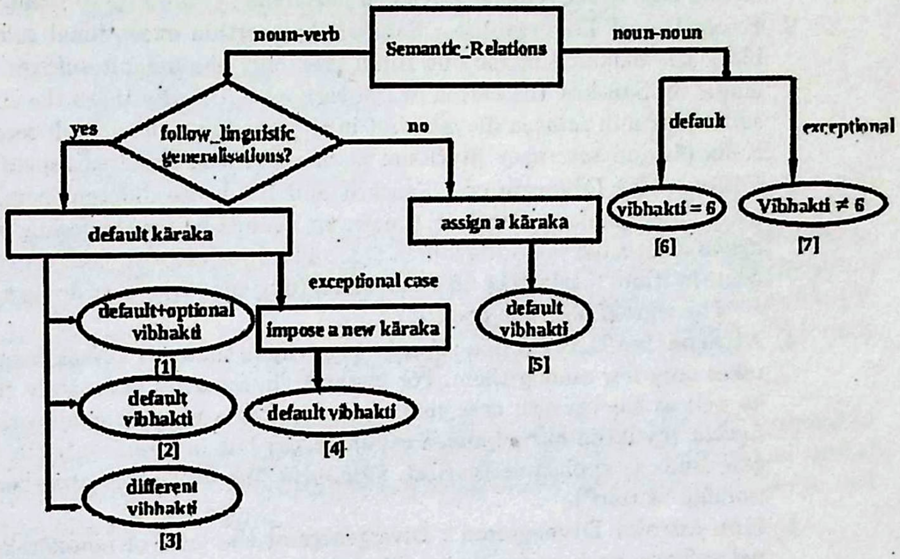
Pāṇini uses kāraka - a syntactico-semantic relation as an intermediary step to express the semantic relations through vibhaktis. The assignment of kāraka to various semantic categories is not one-to-one. Rama Nath Sharma (2002) observes-

> Pāṇini specifies his kāraka categories based upon the principle of sāmānya 'general', viśeṣa 'particular' and śeṣa 'residual'. The six categories are identified by general rules formulated based upon linguistic generalizations. Particular rules form exceptions to them. Usage which cannot be accounted for by the above two rule types is governed by rules relegated to the residual category. It is obvious that these exceptions are necessary to capture the peculiarities of usage falling outside the scope of the general rules.

Each kāraka in his system has a default vibhakti. But as is well-known, there are exceptions and hence there is no one-to-one mapping between the kāraka relations to vibhaktis. Pāṇini handles these deviations by employing two methods: (a) imposing a kāraka role and (b) assigning a special vibhakti. For example through the sūtra *ādhāro'dhikaraṇam* (P-1.4.45) a *locus* is mapped to *adhikaraṇa* and then it takes seventh case suffix by default (*saptamyadhikaraṇe ca* P-2.3.36). However the locus of the action related to the verbal roots *śīṅ, sthā and āsa* preceded by the upasarga *adhi* is termed as karma (*adhiśīṅsthāsāṃ karma* P-1.4.46) and then by the sūtra *karmaṇi dvitīyā* (P-2.3.2) this karma takes the second case suffix. Thus the deviation of the adhikaraṇa taking the second case suffix instead of seventh is handled by imposing a karma kāraka role in place of adhikaraṇa. Similarly, when the linguistic generalisations cannot be captured, he treats the cases as exceptional as in *rucyarthānām prīyamāṇaḥ* (P-1.4.33), where he assigns a sampradāna kāraka to the one who desires. In Fig 1, we summarise Pāṇini's way of mapping semantic relations to vibhaktis through kārakas.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.



1. Divaḥ karma ca (sādhakatamam, kārake) (P–1.4.43)
2. Karmaṇi dvitīyā (P–2.3.2)
3. Divastadarthasya (karmaṇi, ṣaṣṭhī śeṣe, anabhihite) (P–2.3.58)
4. Adhiśīṅsthāsām karma (ādhāraḥ, kārake) (P–1.4.46)
5. Krudhadruhērṣyānām yam prati kopaḥ (sampradānam, kārake) (P–1.4.37)
6. Ṣaṣṭhī śeṣe (P–2.3.50)
7. Pṛthagvinānānābhistṛtīyānyatarasyām (dvitīyā, pañcamī) (P–2.3.32)

Fig. 1. Semantic–Vibhakti_Mapping

This classification does not hold good as-it-is for Hindi. If a parallel grammar for Hindi in the Aṣṭādhyāyī style were available, it would have been a simple task to arrive at the divergence cases. In the absence of such a grammar, the grammar rules corresponding to exceptional cases and cases which can not be captured by linguistic generalisations, were all checked for corresponding vibhaktis in Hindi. Leaving aside those cases where vibhakti is the same in Sanskrit and Hindi and taking up those cases where it diverges, we found that the cases of divergences may be classified into seven types, such as -

1. **Optional Divergence :** Vibhaktis optionally found in Sanskrit but absent in Hindi. For instance, in Sanskrit the karma[2] of the verb is expressed by

[2] Roughly equivalent to the notion of *object* in western linguistics

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

the default second case suffix and optionally the fourth case suffix (bālakāḥ *vidyālayāya/vidyālayaṃ* gacchanti) but in Hindi it takes only the default second case suffix (bālaka *vidyālaya* jāte_haiṁ = 'Boys go to school').

2. **Exceptional Divergence :** Sanskrit has certain exceptional rules which block the default suffixes but Hindi uses only the default suffixes. For example, in Sanskrit the karma of the verb exceptionally takes the sixth case suffix (śakuniḥ *śatasya* dīvyati) but in Hindi it takes the default second case suffix (śakuni *sau_rupae* jītatā_hai = 'Shakuni wins hundred rupees').
3. **Differential Divergence :** Sanskrit and Hindi use different nominal suffixes, for example, in Sanskrit, the person against whom the feeling of hatred *druha* is directed takes the fourth case suffix (durjanāḥ *sajjanāya* druhyanti) while in Hindi it takes the fifth case suffix (durjana *sajjana_se* droha karte_haiṁ = 'The wicked hate the good').
4. **Alternative Divergence :** Sanskrit uses more than one vibhaktis but Hindi takes only few among them. For instance, in Sanskrit alternately the sixth as well as the seventh case suffix is used after a word in conjunction with *āyukta* (āyuktaḥ *haripūjane/haripūjanasya*) but in Hindi only the seventh case suffix is applicable (hari_kī_ *pūjā_meṁ* līna = 'Deeply absorbed in the worship of Hari').
5. **Non-kāraka Divergence :** Divergences at the level of non-kāraka nominal suffixes, such as upapada vibhaktis, sambandha vibhaktis, etc. For instance, in Sanskrit the word takes fifth case suffix when governed by the karmapravachanīya *prati* (pradyumnaḥ *kṛṣṇāt* prati asti) while in Hindi it takes the sixth case suffix (pradyumna *kṛṣṇa_ke* pratinidhi hai = 'Pradyumna is the representative of Krishna').
6. **Verbal Divergence :** Divergences due to the special demand of certain verbs. For example, the karma of the verb *ā+ruh* in Sanskrit (vānaraḥ *vṛkṣam* ārohati) takes the seventh case suffix in Hindi (bandara *peṛa_para* caṛhatā_hai = 'Monkey climbs on the tree').
7. **Complex-Predicate Divergence :** This divergence results when a Sanskrit verb is mapped to a complex predicate in Hindi. For instance, the karma of the verb *anu+sṛ* in Sanskrit (*rāmam* anusarati sītā) is expressed by the genitive case in Hindi (*rāma_kā* sītā anusaraṇa karatī_hai = 'Sita follows Rama').

These cases are elaborated in detail.

2.1 Optional Divergence

When a sūtra assigns optionally two different vibhaktis in Sanskrit but Hindi allows only one vibhakti, we term the resulting divergence as *Optional Divergence.*

Illustration:
divaḥ karma ca (sādhakatamam and kārake) (P-1.4.43) That kāraka which is supplemental in the accomplishment of the action of the verbal root

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

divu (to play) is termed as karma, in addition to karaṇa. e.g.,

(1) San: *akṣān/akṣaiḥ* dīvyati
Hin: *pāsoṁ_se* khelatā_hai
gloss: dice_with plays
Eng: '(He) plays with dice.'

In the above example (1), *akṣa* is the instrument of the verbal root *diva* expressed through the third case suffix derived from the general rule *sādhakatamaṃ karaṇam* (P-1.4.42) but due to the optional rule *divaḥ karma ca*, *akṣa* optionally takes the second case suffix. Whereas in Hindi this optional rule does not apply and thus Hindi allows only the third case suffix.

Table 1 lists the sūtras which account for an additional optional vibhakti in Sanskrit either on account of imposed kāraka (Nos.1-3) or on account of special vibhakti assignment. In all these cases Hindi allows only the default kāraka and hence the default vibhakti.

Sr.No.	Sūtra		Default		Optional	
			Kāraka	Vibhakti	Kāraka	Vibhakti
1	1.4.43	divaḥ karma ca	karaṇa	3	karma	2
2	1.4.44	parikrayaṇe.	karaṇa	3	sampradāna	4
3	1.4.53	hṛkronyatarasyām	kartā	3	karma	2
4	2.3.12	gatyarthakarmaṇi.	karma	2	-	4
5	2.3.22	sañjño'nyatarasyāṃ.	karma	2	-	3
6	2.3.31	enapā dvitīyā	-	6	-	2
7	2.3.59	vibhāṣopasarge	karma	2	-	6
8	2.3.71	kṛtyānāṃ karttari.	kartā	3	-	6

Table 1. *Optional Divergence*

2.2 Exceptional Divergence

When sūtras give exceptional rules for cases by restricting the general rules but these rules are not applicable in Hindi, the divergence is termed as *Exceptional Divergence.*

Illustration:
adhiśīṅsthāsāṃ karma (ādhāraḥ and kārake) (P-1.4.46) That kāraka which is the locus of the verbal roots *śīṅ* (to lie down), *sthā* (to stand), *āsa* (to

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

sit), when preceded by the upasarga *adhi* is termed as karma. e.g.,

(2) San: bālakaḥ *paryaṅkam* adhiśete
Hin: laḍakā *palaṃga_para* sotā_hai
gloss: boy bed_on sleeps
Eng: 'Boy sleeps on the bed'.

In this example, the locus *paryaṅka* of the verbal root *adhi + śīṅ* is termed as karma and thus takes the second case suffix. But this rule does not apply in Hindi. So *paryaṅka* which is the locus and hence adhikaraṇa by the rule *ādhāro'dhikaraṇam* (P-1.4.45) takes the seventh case suffix. This applies to other verbs *sthā and āsa* with prefix adhi as well.

Table 2 lists the sūtras which account for non-default vibhaktis in Sanskrit by imposing a new kāraka (as in Nos.1-3) or by imposing an altogether different vibhakti for the kāraka. In all these cases, Hindi however takes the default kāraka and hence default vibhakti.

Sr.No.	Sūtra		Default		Exceptional	
			Kāraka	Vibhakti	Kāraka	Vibhakti
1	1.4.46	adhiśīṅsthāsāṃ karma	adhikaraṇa	7	karma	2
2	1.4.47	abhiniviśaśca	adhikaraṇa	7	karma	2
3	1.4.48	upānvadhyāṅvasaḥ	adhikaraṇa	7	karma	2
4	2.3.58	divaḥ tadarthasya	karma	2	-	6
5	2.3.61	preṣyabruvorhaviṣo	karma	2	-	6
6	2.3.64	kṛtvo'rthaprayōge	adhikaraṇa	7	-	6

Table 2. *Exceptional Divergence*

2.3 Differential Divergence

All those cases where Pāṇini's rule assigns a certain vibhakti either through kāraka or by direct case assignment and Hindi uses altogether different vibhakti are termed as the case of *Differential Divergence.*

Illustration:
apavargē tṛtīyā (kālādhvanoḥ atyantasaṁyoge) (P-2.3.6) The third case-suffix is employed after the words denoting the duration of time or place (*adhvan*), when accomplishment (*apavarga*) of the desired object is meant to be expressed. e.g.,

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

(3) San: *ahnā* anuvākaḥ adhītaḥ
Hin: *dinabhara_meṁ* anuvāka (veda kā eka bhāga) paṛha_liyā
gloss: whole day_in anuvaka (veda of one part) read
Eng: 'Anuvaka was perseverely and effectually studied (by him) in one day'.

In example (3), the person not only studied *anuvāka* but completely understood and memorised them. Attainment of knowledge is the fruitful result and hence the word *ahnā* which denotes the duration of time takes the third case-suffix. But in Hindi it takes the seventh case-suffix.

Table 3 lists the sūtras which describe cases where Hindi vibhakti differs from the Sanskrit vibhakti.

Sr.No.	Sūtra		Sanskrit		Hindi
			Kāraka	Vibhakti	Vibhakti
1	1.4.34	ślāghahnuṇsthā.	sampradāna	4	2,4,6
2	1.4.35	dhāreruttamarṇaḥ	sampradāna	4	6
3	1.4.37	krudhadruherṣyāsūyā.	sampradāna	4	5,7
4	1.4.38	krudhadruhorupasṛṣṭa.	karma	2	5,7
5	1.4.39	rādhīkṣyoryasya.	sampradāna	4	6
6	2.3.6	apavargē tṛtīyā	-	3	7
7	2.3.17	manyakarmaṇyanādare.	karma	2,4	6
8	2.3.23	hetau	-	3	4
9	2.3.37	yasya ca bhāvena.	-	7	6
10	2.3.43	sādhunipuṇābhyām.	-	7	4
11	2.3.67	ktasya ca varttamāne	-	6	3

Table 3. *Differential Divergence*

2.4 Alternative Divergence

In case of *Alternative Divergence* Sanskrit allows more than one case suffixes and Hindi takes only a few of them, and rarely an altogether different case suffix.

Illustration:
dūrāntikārthaiḥ ṣaṣṭhyanyatarasyām (pañcamī) (P-2.3.34) When in conjunction with *dūra* (distant,far), and *antika* (near) and their synonyms, the sixth case suffix is optionally employed and alternately the fifth. e.g.,

(4) San: *grāmasya/grāmāt* vanaṃ dūram asti

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

Hin: *gānva_se* jaṅgala dūra_hai
gloss: village_from forest far is
Eng: 'Forest is far from the village'.

In example (4), in Sanskrit *grāma* takes optionally the sixth case suffix as well as the fifth when combined with *dūra* as well as *antika* but Hindi takes only the fifth case suffix.

Table 4 lists the sūtras where Sanskrit takes more than one vibhaktis but Hindi takes only a few of them.

Sr.No.	Sūtra		Sanskrit	Hindi
			Vibhakti	Vibhakti
1	2.3.32	pṛthagvinānānā.	2,3,5	5,6
2	2.3.34	dūrāntikārthaiḥ.	5,6	5
3	2.3.36	saptamyadhikaraṇe ca	2,3,5,7	0
4	2.3.39	svāmīśvarādhipati.	6,7	6
5	2.3.40	āyuktakuśalābhyāṃ.	6,7	7
6	2.3.41	yataśca nirdhāraṇam	6,7	7
7	2.3.44	prasitotsukābhyāṁ.	3,7	2,7
8	2.3.72	tulyārthairatulopamābhyāṁ.	3,6	6
9	2.3.73	caturthī cāśiṣyāyuṣya.	4,6	2,6

Table 4. *Alternative Divergence*

2.5 Non-kāraka Divergence

Certain words known as *upapada* demand specific vibhaktis called *Upapada Vibhaktis* for nouns with which they combine. These are all typically exceptions to the *ṣaṣṭhī śeṣe* (P-2.3.50). In Hindi, however, in most of the cases sixth case suffix is used.

Illustration:
sahayukte apradhāne (tṛtīyā) (P-2.3.19) The word denoting *apradhāna* in conjunction with *saha* and its synonyms takes the third case suffix. e.g.,

(5) San: *rāmeṇa* saha sītā vanaṁ gacchati
Hin: *rāma_ke* sātha sītā vana jātī hai
gloss: rama_with sita forest goes
Eng: 'Sita goes to forest with Rama'.

CC0. In Public Domain Kṛtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

In example (5), in Sanskrit it is seen that *sītā and vana* are directly related to the verb in the form of kartā and karma but *rāma* is not directly related to the verb and hence is called *apradhāna*. The association of *rāma* with *sītā* is denoted by the word *saha* due to which *rāma* takes the third case suffix from the above rule. But in the case of Hindi, *rāma* in conjunction with *saha* takes only the sixth case suffix.

Table 5 lists the sūtras where Sanskrit generally takes the second case suffix barring the sixth case suffix but Hindi does not take this exemption into account and takes the sixth case suffix.

Sr.No.	Sūtra		Sanskrit Vibhakti	Hindi Vibhakti
1	1.4.84	tṛtīyārthe	2	6
2	1.4.85	hine	2	6
3	1.4.87	upo'dhike ca	2	6
4	1.4.89	lakṣaṇetthambhūtā.	2	6
5	1.4.90	abhirabhāge	2	6
6	1.4.95	atiratikramaṇe ca	2	5,6
7	2.3.4	antarā'ntareṇayukte	2	6
8	2.3.5	kālādhvanoratyantasanyoge	2	7
9	2.3.10	pañcamyapāṅparibhiḥ	5	2
10	2.3.11	pratinidhipratidāne ca yasmāt	5	6
11	2.3.16	namaḥ svastisvāhā.	4	2,4,6
12	2.3.19	sahayukte apradhāne	3	6
13	2.3.26	anyārāditarartedik.	5	5,6

Table 5. *Non-kāraka Divergence*

2.6 Hindi Specific Divergences

All the divergences we covered so far were specific to the Aṣṭādhyāyī, and some of them might be attributed to the characteristics of Sanskrit. The default vibhakti of karma in Sanskrit is dvitīyā. But in Hindi, the karma takes vibhakti other than dvitīyā. These divergences may be attributed to the nature of Hindi. On the face of it divergences may be classified into two classes depending upon whether the Sanskrit verb maps to a simple Hindi verb or a complex predicate.

(a) Verbal Divergence
Consider the example,

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

(6) San: vānaraḥ *vṛkṣam* ārohati
Hin: bandara *peṛa_para* caṛhatā_hai
gloss: monkey tree_on climbs
Eng: 'Monkey climbs on the tree'.

Here, the karma of the verbal root *ā + ruh* is *vṛkṣa* whereas in Hindi the karma *vṛkṣa* takes seventh case suffix. The important question from Hindi grammar point of view then is - whether vṛkṣa should be termed as a karma as in Sanskrit or should it be termed as an adhikaraṇa? If we term it as a karma, then the treatment will be close to Sanskrit but one needs to impose a rule accounting for the divergence in the suffix. If we term it as an adhikaraṇa, then we deviate from the Sanskrit grammar but can take an advantage of the default vibhakti mapping.
There are many such instances, and these should be treated carefully by Sanskrit scholars who know Hindi well, taking insights from how Pāṇini handled such cases.

(b) Complex Predicate Divergence
If a verb in Sanskrit maps to a complex predicate[3] in Hindi, then the karma of the verb takes the sixth case suffix. e.g.,

(7) San: *rāmam* anusarati sītā
Hin: *rāma_kā* sītā anusaraṇa karatī_hai
gloss: rama_of sita follows
Eng: 'Sita follows Rama'.

Here in example (7), *rāma* which is the karma is expressed with the genitive case in Hindi. This change is a systematic one which may be explained through the rule *kartṛkarmaṇoḥ kṛti* (P-2.3.65). In Hindi *anu + sṛ* is translated as a complex predicate *anusaraṇa_karanā* where *anusaraṇa* is the kṛdanta and *karanā* is the main verb.

It is necessary to study these divergences in karma vibhakti in detail. Sanskrit has approximately 2000 verbal roots and many more with the *upasargas*. Mapping the corresponding verb-frames in Sanskrit to the Hindi verb-frames may give some hints towards these divergences. Though it is a voluminous task, any MT system requires this study.

3 Conclusion

The cases of divergences may be summarised then as:
(A) Divergences originating from Sanskrit Grammar (as shown below)

[3] A complex predicate consists of a noun followed by a light verb such as karanā, honā, etc. e.g. vivāha karanā, snāna karanā, etc.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

Sr.No.	Cases	Sanskrit	Hindi
1	Optional	Sanskrit uses optional vibhakti in addition to default vibhakti.	Hindi allows only the default vibhakti.
2	Exceptional	Sanskrit uses different vibhakti than the default (Pāṇini blocks the default vibhakti by treating it as an exceptional case).	Hindi uses the default vibhakti.
3	Differential	Pāṇini imposes certain kārakas or vibhaktis which cannot be captured through the semantic generalisations.	Hindi uses different vibhaktis.
4	Alternative	Sanskrit uses more than one alternative vibhakti.	Hindi uses only one of them.
5	Non-kāraka	Sanskrit uses more than one alternative vibhakti.	Hindi uses the sixth vibhakti.

(B) Divergences due to the idiosyncrasy of Hindi:

1. Special vibhakti expectancy of verbs
2. Complex Predicate

These seven cases of divergence may prove useful in resolving the ambiguities at the level of nominal suffixes in Machine Translation and may serve as a platform for looking at vibhakti level divergences between Sanskrit and other Modern Indian Languages.

Acknowledgements
This work is a part of the Sanskrit Consortium project entitled 'Development of Sanskrit computational tools and Sanskrit-Hindi Machine Translation system' sponsored by the Government of India. We gratefully acknowledge Prof.Lakshmi Bai for the discussions at various stages.

References

1. Acharya Ramshastri: *Saṁskṛta śikṣaṇa saraṇī.* Acharya Ramshastri Jñānapīṭha (Hindi) (1998)
2. Cardoṇa, George: *Pāṇini and Pāṇinīyas on śeṣa Relations.* Kunjunni Raja Academy of Indological Research, Kochi (2007)
3. Dorr, Bonnie: *Classification of Machine Translation Divergence and a Proposed Solution.* Computational Linguistics 20(4) (1994)
4. Dave, S., Parikh, J., Bhattacharya, P.: *Interlingua Based English-Hindi Machine Translation and Language Divergence.* Journal of Machine Translation (JMT) 17 (2002)
5. Dwivedi, Kapil Deva: *Bhāṣā vijñāna evam Bhāṣā śāstra.* Viśvavidyālaya Prakāśana, Varanasi (2006)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

6. Goyal, Pawan and Sinha, R. Mahesh K.: *Translation Divergence in English-Sanskrit-Hindi Language Pairs*. In Amba Kulkarni, and Gérard Huet, editors, Sanskrit Computational Linguistics, Proceedings of Third International Symposium, Hyderabad. Springer-Verlag LNAI 5406 (2009)
7. Mishra, Vimal and Mishra, R. B.: *Study of Example Based English to Sanskrit Machine Translation*. Journal of Research and Development in Comp Sc. And Engg Number 37, Jan-June (2008)
8. Sharma, Ram Rang and Sharma, Malti: *Vaiyākaraṇa Siddhānta Kaumudī of Śrī Bhaṭṭoji Dīkṣita (Kāraka Prakaraṇam)*. Bharatiya Vidya Prakashan, Varanasi-Delhi (Hindi) (1997)
9. Sharma, Rama Nath: *The Aṣṭādhyāyī of Pāṇini*. Vols 1-3. Munshiram Manoharlal, Delhi 2nd Ed (2002)
10. Vasu, Śrīśa Chandra: *The Siddhānta Kaumudī of Bhaṭṭoji Dīkṣita*. Vol I. Motilal Banarsidass, Delhi (2003)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

संस्कृत -हिन्दीभाषयोः विभक्ति-व्यत्यासः

प्रीति शुक्ला, देवानन्द शुक्ल, अम्बा कुलकर्णी
संस्कृताध्ययनविभागः, हैदराबाद-विश्वविद्यालयः
भाग्यनगरम् (हैदराबाद) भारतम्
shukla.preetidev@gmail.com, dev.shukl@gmail.com, apksh@uohyd.ernet.in

सारांशः अनुवादपरम्परायां विभिन्नभाषाणां मध्ये व्यत्यासः विभिन्नस्तरेषु दरीदृश्यते। स व्यत्यासः व्याकरणाधारितः भवति उत तस्य अन्यानि कारणान्यपि भवितुमर्हन्ति। संस्कृतम् एवं हिन्दी उभेऽपि भारोपीयपरिवारीये स्तः। संरचनायां एवञ्च शब्दस्तरे संस्कृतभाषातः हिन्दीभाषायां यथावत् बहवः अंशाः आगताः। तथापि तत्र विभक्तिदृष्ट्या व्यत्यासः दृश्यते। अष्टाध्याय्यां पाणिनिना अपवादान् स्वीकृत्य एव उत्सर्गतः कारकाकेषु विभक्तिनिर्धारणं कृतम्। अपवादेषु नूतनकारकस्यारोपः कृतः उत तत्र विशिष्टविभक्तिनिर्धारणं कृतम्। परन्तु एषा प्रक्रिया हिन्दीभाषायां पूर्णतया स्वीकरणीया न भवति। एतान् सप्तविधान् व्यत्यासानाधारीकृत्य शोधपत्रमिदं लिखितमस्माभिः।

परिचयः

विभिन्नभाषाः व्याकरणाधारीकृत्य सूचनां प्रस्तुतीकरणार्थं विभिन्न-प्रक्रियाः स्वीकुर्वन्ति। येन सामान्यतः अनुवादे तथा च यान्त्रिकानुवादे विशेषतः विभिन्नस्तरेषु भाषायां व्यत्यासाः दृश्यन्ते। डोर (Dorr, 1994) इति नाम्नी विदुषी अनुवाद-व्यत्यासविषयकप्रक्रिया-विषये चर्चां अकरोत्। तां प्रक्रियामाधारीकृत्य भारतीयभाषासु कार्याण्यपि जातानि। यथा आंग्ल-संस्कृत-हिन्दी-यान्त्रिकानुवादः[1], आंग्ल-संस्कृत-यान्त्रिकानुवादः[2], आंग्ल-हिन्दी-यान्त्रिकानुवादः[3].

संस्कृतम् एवं हिन्दी उभेऽपि भारोपीयपरिवारीये स्तः। संरचनायां एवञ्च शब्दस्तरे संस्कृतभाषातः हिन्दीभाषायां यथावत् बहवः अंशाः आगताः। यदा वयं संस्कृत-हिन्दी-मध्ये व्यत्यासं पश्यामः तदा प्रायः डोर इत्यनया प्रतिपादिते यद्व्यत्यासे यथाकथञ्चिदेव दृश्यते अथवा यान्त्रिकसाधनैः मूलस्रोतस्वीकरणे क्लेशं नोत्पादयन्ति। किञ्च संस्कृत-हिन्दीभाषयोः व्यत्यासाः विभक्तिस्तरे विविधरूपेण दृश्यन्ते। संस्कृतम् एवञ्च हिन्दी उभेऽपि श्लिष्टभाषे स्तः परन्तु संस्कृतं संयोगात्मिका भाषा तथा हिन्दी वियोगात्मिका भाषा[4]। संस्कृतभाषायां सुप्तिङौ विभक्तिसंज्ञौ स्तः। सुप्प्रत्ययाः प्रातिपदिकात् भवन्ति, तिङ् धातोः। परन्तु हिन्दीभाषायां प्रातिपदिकेभ्यः ये प्रत्ययाः भवन्ति ते एव विभक्तिसंज्ञकाः। शोधपत्रेऽस्मिन् संस्कृत-हिन्दी-विभक्तिव्यत्यास-विषये चर्चां कुर्मः।

1 English-Sanskrit-Hindi MT by Goyal and Sinha, 2009.
2 English-Sanskrit MT by Mishra and Mishra, 2008.
3 English-Hindi MT by Dave et al., 2002.
4 द्विवेदी, 2006

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

व्यत्यासानां वर्गीकरणम्

विभक्त्या आर्थिकपक्षस्य विवरणार्थं पाणिनिः कारकस्योपयोगं करोति। आर्थिकदृष्ट्या प्रत्येकस्य कारकस्य कृते एका एव विभक्तिः नोपयुज्यते। एतस्य विषये पं. रमानाथशर्मणा (2002) स्वकीये व्याख्याने लिखितम् –

पाणिनिः कारकस्य विभाजनं सैद्धान्तिकदृष्ट्या 'सामान्यः विशेषः एवं शेषः' इत्याधारेण करोति। "पारिभाषिक–आरोपित–कारकनियमाः" इत्याधारेण षट्कारकाणां कृते सामान्यनियमाः सन्ति। विशेषनियमाः तेषाम् अपवादस्वरूपाः सन्ति। येषां सामान्य–विशेषयोरन्तर्भावः न भवति ते शेषनियमाः भवन्ति। सामान्यनियमेतरगुणानामुपयोगदृष्ट्या अपवादाः आवश्यकाः।

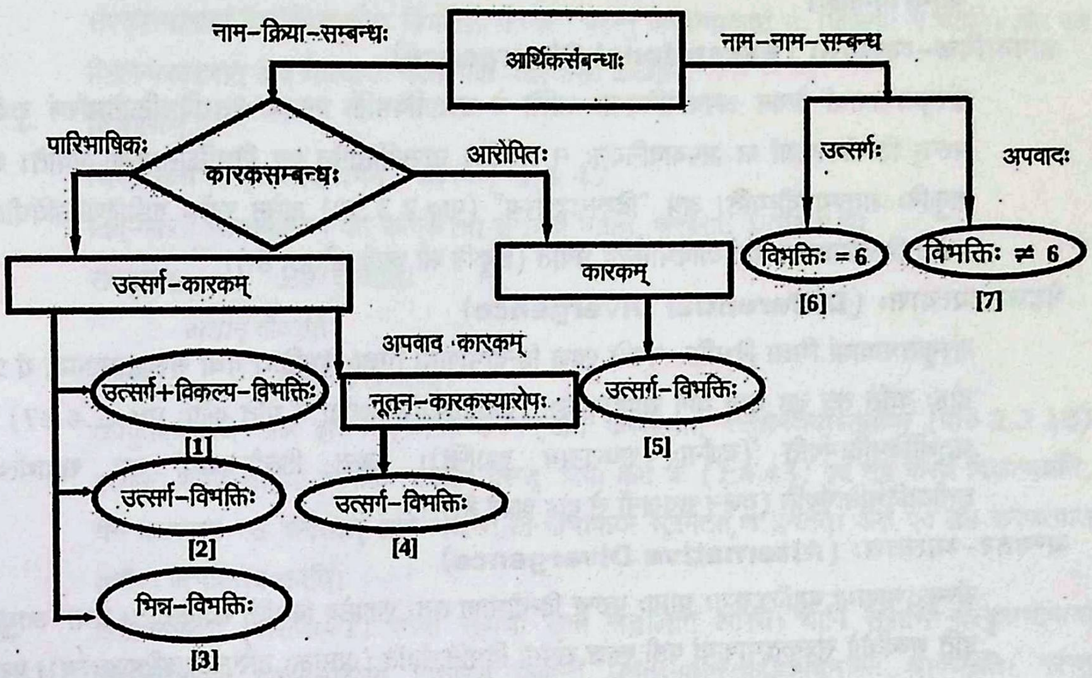
(Pāṇini specifies his kāraka categories based upon the principle of sāmānya 'general', viśeṣa 'particular' and śeṣa'residual'. The six categories are identified by general rules formulated based upon linguistic generalizations. Particular rules form exceptions to them. Usage which cannot be accounted for by the above two rule types is governed by rules relegated to the residual category. It is obvious that these exceptions are necessary to capture the peculiarities of usage falling outside the scope of the general rules.)

पाणिनीयप्रक्रियायां प्रत्येकस्य कारकस्य कृते एका विभक्तिः निर्धारिता। परन्तु तत्र अपवादाः अपि सन्ति यस्य कारणं कारकविभक्त्योः मध्ये निश्चितरूपेण एकैकस्य कारकस्य कृते एकैव विभक्तिः न निर्धारिता। पाणिनिः द्विधा एतस्य निराकरणं करोति। (अ) कारकस्यारोपं कृत्वा (ब) विशेषविभक्तिनिर्धारणं कृत्वा।

यथा "आधारोऽधिकरणम्" (पा० 1.4.45) सूत्रमिदं आधारस्य अधिकरणसंज्ञां करोति। अनन्तरं तस्मिन्नधिकरणे सामान्यसूत्रेण (सप्तम्यधिकरणे च पा० 2.3.36) सप्तमीविभक्तिर्भवति। परन्तु यदि आधारः अधि पूर्वकः शीङ्, स्था, आस् इत्यस्मात् धातोः सम्बद्धः अस्ति चेत् तस्य कर्मसंज्ञा (अधिशीङ्स्थासां कर्म पा० 1.4.46) भवति एवं च "कर्मणि द्वितीया" (पा० 2.3.2) इत्यनेन कर्मणि द्वितीया विभक्तिर्विधीयते। उदाहरणेऽस्मिन् पाणिनिना सप्तमीविभक्तिवारणाय नूतनकारकस्यारोपं कृत्वा आधारे कर्मसंज्ञामारोप्य तत्र द्वितीया विभक्तिः निर्धार्यते। अनेन प्रकारणैव यत्र पारिभाषिकं कारकं न भवति तत्र साक्षात् कारकमारोपयति। यथा "रुच्यर्थानां प्रीयमाणः" (पा० 1.4.33) रुच्यर्थानां धातूनां प्रयोगे यः प्रीयमाणः सः सम्प्रदानसंज्ञकः भवति। अग्रिमचित्रे आर्थिकदृष्ट्या कारक–विभक्ति–सम्बन्धः प्रदर्श्यते सामान्यरूपेण –

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri



1. दिवः कर्म च (साधकतमम, कारके) (पा० 1.4.43)
2. कर्मणि द्वितीया (पा० 2.3.2)
3. दिवस्तदर्थस्य (कर्मणि, षष्ठी शेषे, अनभिहिते) (पा० 2.3.58)
4. अधिशीङ्स्थासां कर्म (आधारः, कारके) (पा० 1.4.46)
5. क्रुधद्रुहेर्ष्यसूयार्थानां यं प्रति कोपः (सम्प्रदानम्, कारके) (पा० 1.4.37)
6. षष्ठी शेषे (पा० 2.3.50)
7. पृथग्विनानानाभिस्तृतीयान्यतरस्याम् (द्वितीया, पञ्चमी) (पा० 2.3.32)

एतादृशं विवरणात्मकं वर्गीकरणं हिन्दीभाषायां नोपयुज्यते। यदि हिन्दीभाषायाम् अष्टाध्यायीसदृशं व्याकरणमभविष्यत् तर्हि व्यत्यासानां निराकरणं सरलं भविष्यति। तादृशव्याकरणाभावात् अपवाददृष्ट्या व्याकरणस्य नियमानाम् एवञ्च पारिभाषिकारोपितकारकाभ्यां नान्तर्भवन्ति तान् नियमानां परीक्षणं कृत्वा संस्कृत–हिन्दीभाषयोः व्यत्यासः प्रदर्शितः। सः व्यत्यासः सप्तप्रकारकः भवति –

वैकल्पिक–व्यत्यासः (Optional Divergence)

संस्कृतभाषायां वैकल्पिकरूपेण विभक्ती द्वे प्राप्येते। परन्तु हिन्दीभाषाया सः विकल्पः न भवति। यथा संस्कृतभाषायां 'बालकः विद्यालयं गच्छति' इत्यत्र सामान्यतया 'विद्यालय' इत्यस्य कर्मसंज्ञा भवति तस्य फलं 'विद्यालय' इत्यत्र द्वितीयाविभक्तिः आगच्छति। परन्तु अत्रैव वैकल्पिकत्वेन "गत्यर्थकर्मणिद्वितीयाचतुर्थ्यौ चेष्टायामनध्वनि" (2.3.12) अनेन सूत्रेण कर्मणि चतुर्थी विभक्तिरपि विधीयते। तस्य फलं 'विद्यालय' इत्यत्र चतुर्थी विभक्तिर्भवति। 'बालकः विद्यालयाय गच्छति' इति रूपं

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

भवति। अपरत्र हिन्दीभाषायां कर्मत्वेन द्वितीयाविभक्तिरेव भवति (बालक विद्यालय जाता है।) न तु चतुर्थी विभक्तिः।

आपवादिक-व्यत्यासः (Exceptional Divergence)

संस्कृतभाषायां केचन अपवादनियमाः सन्ति ये उत्सर्गविभक्तिं प्रबाध्य इतरविभक्तिनिर्धारणं कुर्वन्ति परन्तु हिन्दीभाषायां स अपवादनियमः न प्रवर्तते। उत्सर्गनियमेन एव विभक्तिनिर्धारणं भवति। यथा शकुनिः शतस्य दीव्यति। अत्र "दिवस्तदर्थस्य" (पा० 2.3.58) अनेन सूत्रेण षष्ठीविभक्तिर्विधीयते। परन्तु हिन्दीभाषायां द्वितीयाविभक्तिरेव भवति (शकुनि सौ रुपये जीतता है।)।

भेदज-व्यत्यासः (Differential Divergence)

संस्कृतभाषायां भिन्ना विभक्तिः भवति एवञ्च हिन्दीभाषायां भिन्ना विभक्तिः। यथा संस्कृतभाषायां यं प्रति द्रोहः वर्तते तत्र द्रुह धातु योगे संप्रदानसंज्ञा (क्रुधद्रुहेर्ष्यासूयार्थानां यं प्रति कोपः पा० 1.4.37) एवं चतुर्थीविभक्तिर्भवति (दुर्जनाः सज्जनाय द्रुह्यन्ति)। परन्तु हिन्दीभाषायां तत्र सहार्थत्वात् तृतीयाविभक्तिर्भवति (दुर्जन सज्जनों से द्रोह करते हैं।)

अन्यतर-व्यत्यासः (Alternative Divergence)

संस्कृतभाषायां बहुविभक्तयः प्राप्ताः परन्तु हिन्दीभाषा ततः एकामेव विभक्तिं स्वीकरोति। यथा 'आयुक्त' इति शब्दयोगे संस्कृतभाषायां षष्ठी एवञ्च सप्तमी विभक्तिर्भवति (आयुक्तः हरिपूजने/हरिपूजनस्य)। परन्तु हिन्दीभाषायां सप्तमीविभक्तिरेव भवति (हरि की पूजा में लीन।)

अकारक-व्यत्यासः (Non-Karaka Divergence)

कारकेतरविभक्तयः उपपदविभक्तयः सम्बन्धद्योतिकाः विभक्तयः भवन्ति। ताः आधारीकृत्य यः व्यत्यासः सः अकारक-व्यत्यासः। यथा 'प्रति' एतस्य यदा कर्मप्रवचीयसंज्ञा भवति तदा तत्र पञ्चमीविभक्तिरुपयुज्यते (प्रद्युम्नः कृष्णात् प्रति अस्ति)। परन्तु हिन्दीभाषायां तत्र सम्बन्धद्योतिका षष्ठीविभक्तिरेव भवति (प्रद्युम्न कृष्ण का प्रतिनिधि है।)

क्रियास्वरूप-व्यत्यासः (Verbal Divergence)

क्रियामाधारीकृत्य व्यत्यासः भवति। यथा संस्कृतभाषायाम् 'आ+रुह" धातोः यः कर्म तत्र द्वितीयाविभक्तिर्भवति (वानरः वृक्षम् आरोहति।) परन्तु हिन्दीभाषायां 'चढ' इति धातुः अकर्मकः धातुः अतः आधारत्वेन विवक्षितत्वात् सप्तमीविभक्तिर्भवति (बन्दर पेड पर चढता है।)।

क्रियामूल-व्यत्यासः (Complex-Predicate Divergence)

संस्कृतभाषायां यत्र क्रिया एका भवति परन्तु हिन्दीभाषायां तस्याः परिवर्तनं क्रियामूलरूपेण भवति। यथा संस्कृतभाषायां 'अनु+सृ' इति धातोः यत् कर्म तत्र द्वितीया विभक्तिर्वर्तते (सीता रामम् अनुसरति।) परन्तु अयं धातुः यदा हिन्दीभाषायां 'अनुसरण कर' इति रूपेण परिवर्त्यते तदा तत्र कर्मणि षष्ठीविभक्तिः आगच्छति (सीता राम का अनुसरण करती है।)।

इदानीं व्यत्यासानां सविस्तरं विवरणं प्रस्तूयते

वैकल्पिक-व्यत्यासः (Optional Divergence)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

संस्कृतभाषायां वैकल्पिकरूपेण विभक्तिद्वे प्राप्येते। परन्तु हिन्दीभाषायां सः विकल्पः न भवति। अत एव विकल्पकारणात् अयं व्यत्यासः वैकल्पिक-व्यत्यासः कथ्यते।

विवरणम्

दिवः कर्म च (साधकतमम्, कारके) 1.4.43

दिव्-धातोः साधकतममं यत् कारकं तत् कर्मसंज्ञं भवति, चकारात् करणसंज्ञं च।

संस्कृतम् *अक्षैः* दीव्यति।

अक्षान् दीव्यति।

हिन्दी *पासों से* खेलताहै।

उपर्युक्तोदाहरणे 'अक्ष' इति दिव् धातोः करणम् अतः करणत्वात् 'कर्तृकरणयोस्तृतीया' (पा० 2.3.18) इत्यनेन सूत्रेण तृतीया विभक्तिः भवति। परन्तु 'दिवः कर्म च' (1.4.43) इदं सूत्रं करणं विकल्पयति, येन करणस्य पक्षे कर्मसंज्ञा अपि भवति। हिन्दीभाषायां सूत्रमेतत् न प्रवर्तते। अत एव तत्र करणत्वात् तृतीया विभक्तिरेव भवति।

अग्रिमसरण्यां (सरणिः-1) एतेषां सूत्राणां सूची सङ्कलिता अस्ति। यानि सूत्राणि संस्कृतभाषायां वैकल्पिकत्वेन इतरकारकस्य निर्धारणं कुर्वन्ति येनेतरकारकत्वाद्भिन्नविभक्तिः आगच्छति। परन्तु हिन्दीभाषा तत्र मूलकारकम् (संस्कृतभाषायां यस्य विकल्पः भवति) एव अङ्गीकरोति। अतः विभक्तिपरिवर्तनस्यापेक्षा नास्ति।

क्रमाङ्कः	सूत्रसङ्ख्या	सूत्रम्	उत्सर्गः		विकल्पः	
			कारकम्	विभक्तिः	कारकम्	विभक्तिः
1	1.4.43	दिवः कर्म च	करण	3	कर्म	2
2	1.4.44	परिक्रयणे सम्प्रदानमन्यतरस्याम्	करण	3	सम्प्रदानम्	4
3	1.4.53	हृकोरन्यतरस्याम्	कर्ता	3	कर्म	4
4	2.3.12	गत्यर्थकर्मणि द्वितीयाचतुर्थ्यौ चेष्टायामनध्वनि	कर्म	2	–	4
5	2.3.22	संज्ञोऽन्यतरस्याम् कर्मणि	कर्म	2	–	3
6	2.3.31	एनपा द्वितीया	–	6	–	2
7	2.3.59	विभाषोपसर्गे	कर्म	2	–	6
8	2.3.71	कृत्यानां कर्तरि	कर्ता	3	–	6

सरणिः **-1** वैकल्पिक-व्यत्यासः

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

आपवादिक-व्यत्यासः (Exceptional Divergence)

संस्कृतभाषायां केचन अपवादनियमाः सन्ति ये उत्सर्गविभक्तिं प्रबाध्य इतरविभक्तिनिर्धारणं कुर्वन्ति परन्तु हिन्दीभाषायां स अपवादनियमः न प्रवर्तते। उत्सर्गनियमेन एव विभक्तिनिर्धारणं भवति। संस्कृतभाषायां अपवादः प्रवर्तते हिन्दीभाषायां अपवादस्यास्याभावः भवति। अतः अपवादकारणात् व्यत्यासोऽयं 'आपवादिक-व्यत्यासः' जातः।

विवरणम्

अधिशीङ्स्थासां कर्म (आधारः, कारके)(पा० 1-4-46)

अधिपूर्वाणां शीङ् स्था आस् इत्येतेषाम् आधारः यः तत् कारकं कर्मसंज्ञं भवति।

संस्कृतम् बालकः *पर्यङ्कम्* अधिशेते।

हिन्दी बालक *पलंग पर* सोता है।

उदाहरणेऽस्मिन् अधि+शीङ् धातोः कर्म 'पर्यङ्क' इत्यस्ति तस्मात् तत्र द्वितीया विभक्तिः। हिन्दीभाषायां सूत्रमिदं न प्रवर्तते। अतः 'पर्यङ्क' इत्यस्य आधारोऽधिकरणम्(पा० 1.4.45) इत्यनेन सूत्रेण अधिकरणसंज्ञा एवञ्च सप्तम्यधिकरणे (पा० 2.3.36) इत्यनेन सूत्रेण तत्र सप्तमी विभक्तिर्भवति। अत्र शीङ् धातोः उदाहरणम् प्रदत्तम् अनेन प्रकारेण अधि+स्था, अधि+आस् इत्यत्रापि ज्ञेयम्।

अग्रिमसरण्यां (सरणिः-2) एतेषां सूत्राणां सूची सङ्कलिता अस्ति। यत्र सामान्यकारकम् प्रबाध्य नूतनकारकस्यारोपः कृतः पाणिनिना। तस्य फलस्वरूपेण तत्र भिन्ना विभक्तिः आगच्छति । परन्तु हिन्दीभाषायां एते नियमाः न प्रवर्तन्ते अतः सामान्यकारकेणैव सामान्यविभक्तिः आगच्छति।

क्रमाङ्कः	सूत्रसङ्ख्या	सूत्रम्	उत्सर्गः		अपवादः	
			कारकम्	विभक्तिः	कारकम्	विभक्तिः
1	1.4.46	अधिशीङ्स्थासां कर्म	अधिकरणम्	7	कर्म	2
2	1.4.47	अभिनिविशश्च	अधिकरणम्	7	कर्म	2
3	1.4.48	उपान्वन्ध्याङ्वसः	अधिकरणम्	7	कर्म	2
4	2.3.58	दिवस्तदर्थस्य	कर्म	2	–	6
5	2.3.61	प्रेष्यब्रुवोर्हविषो देवता सम्प्रदानस्य	कर्म	2	–	6
6	2.3.64	कृत्वोऽर्थप्रयोगे कालेऽधिकरणे	अधिकरणम्	7	–	6

सरणिः – 2 आपवादिक-व्यत्यासः

भेदज-व्यत्यासः (Differential Divergence)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

संस्कृतभाषायां भिन्ना विभक्तिः भवति एवञ्च हिन्दीभाषायां भिन्ना विभक्तिः। अत्र द्वयोः भाषयोः या विभक्तिर्भवति सा पूर्णरूपेण परस्परभिन्नैव भवति। अतः भेदकारणात् अयं व्यत्यासः 'भेदजव्यत्यासः' इति कथ्यते।

विवरणम्

अपवर्गे तृतीया (कालाध्वनोरत्यन्तसंयोगे)(पा० 2-3-6)

अपवर्गे गम्यमाने कालाध्वनोरत्यन्तसंयोगे तृतीया विभक्तिर्भवति।

संस्कृतम् अह्ना अनुवाकः अधीतः।

हिन्दी दिन भर में अनुवाक (वेद का एक भाग) पढ लिया।

उपर्युक्तोदाहरणे अह्ना अनुवाकः अधीतः इत्यस्य तात्पर्यं तेन अध्ययनस्य फलमपि प्राप्तम्। अपवर्गः फलप्राप्तिः, अतः फलप्राप्तिः जाते सति तत्र तृतीया विभक्ति भवति। परन्तु हिन्दीभाषायां एतादृशेषु स्थलेषु नैरन्तर्यं विवक्षितत्वात् तत्र सप्तमीविभक्तिर्भवति।

अग्रिमसरण्यां (सरणिः-3) एतादृशानां सूत्राणां सङ्कलनं कृतम् यत्र संस्कृतभाषायां भिन्ना विभक्तिर्वर्तते एवञ्च हिन्दीभाषायां भिन्ना विभक्तिः।

क्रमाङ्कः	सूत्रसङ्ख्या	सूत्रम्	संस्कृतम्		हिन्दी
			कारकम्	विभक्तिः	विभक्तिः
1	1.4.34	श्लाघह्नुङ्स्थाशपां ज्ञीप्स्यमानः	सम्प्रदानम्	4	2,4,6
2	1.4.35	धारेरुत्तमर्णः	सम्प्रदानम्	4	6
3	1.4.37	क्रुधद्रुहेर्ष्यासूयार्थानां यं प्रति कोपः	सम्प्रदानम्	4	4,7
4	1.4.38	क्रुधद्रुहोरुपसृष्टयोः कर्म	कर्म	2	5,7
5	1.4.39	राधीक्ष्योर्यस्य विप्रश्नः	सम्प्रदान	4	6
6	2.3.6	अपवर्गे तृतीया	–	3	7
7	2.3.17	मन्यकर्मण्यनादरे विभाषाऽप्राणिषु	कर्म	2,4	6
8	2.3.23	हेतौ	–	3	4
9	2.3..37	यस्य च भावेन भावलक्षणम्	–	7	6
10	2.3.43	साधुनिपुणाभ्यामर्चायां सप्तम्यप्रतेः	–	7	4
11	2.3.67	क्तस्य च वर्तमाने	–	6	3

सरणिः **-3** भेदज-व्यत्यासः

अन्यतर-व्यत्यासः (**Alternative Divergence**)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

संस्कृतभाषायां बहुविभक्तयः प्राप्ताः परन्तु हिन्दीभाषा ततः एकामेव विभक्तिं स्वीकरोति। बहूनाम् एक एव विभक्तिस्वीकरणार्थं एषः व्यत्यासः "अन्यतरव्यत्यासः" इति संज्ञां प्राप्नोति।

विवरणम्

दूरान्तिकार्थैः षष्ठ्यन्यतरस्याम् (पञ्चमी) (पा० 2-3-34)

दूरार्थः अन्तिकार्थैः=समीपार्थैः शब्दैः योगे षष्ठीविभक्तिः विकल्पेन भवति, पक्षे पञ्चमी च।

संस्कृतम् ग्रामस्य / ग्रामाद् वनं दूरम् अस्ति।

हिन्दी गाँव से जंगल दूर है।

उपर्युक्तोदाहरणे संस्कृतभाषायां 'दूर' इति शब्दयोगे 'ग्राम' शब्दात् पञ्चमी एवं षष्ठी इति द्वे विभक्ती आगच्छतः। परन्तु हिन्दीभाषायां पञ्चमीविभक्तिरेव भवति।

अग्रिमसरण्यां (सरणिः-4) एतादृशानां सूत्राणां सङ्कलनं कृतम् यत्र संस्कृतभाषायां नैकाः विभक्तीः भवन्ति परन्तु हिन्दीभाषा तासां विभक्तीनामेका एव विभक्तिः स्वीकरोति।

क्रमाङ्कः	सूत्रसङ्ख्या	सूत्रम्	संस्कृतम् विभक्तिः	हिन्दी विभक्तिः
1	2.3.32	पृथग्विनानानाभिस्तृतीयान्यतरस्याम्	2,3,5	5,6
2	2.3.34	दूरान्तिकार्थेभ्यो द्वितीयान्यतरस्याम्	5,6	5
3	2.3.36	सप्तम्यधिकरणे च	2,3,5,7	0
4	2.3.39	स्वामीश्वरादिपतिदायादसाक्षिप्रतिभूप्रसूतैश्च	6,7	6
5	2.3.40	आयुक्तकुशलाभ्यां चासेवायाम्	6,7	7
6	2.3.41	यतश्च निर्धारणम्	6,7	7
7	2.3.44	प्रसितोत्सुकाभ्यां तृतीयान्यतरस्याम्	3,7	2,7
8	2.3.72	तुल्यार्थैरतुलोपमाभ्यां तृतीयान्यतरस्याम्	3,6	6
9	2.3.73	चतुर्थी चाशिष्यायुष्यमद्रभद्रकुशलसुखार्थहितैः	4,6	2,6

सरणिः -4 अन्यतर-व्यत्यासः

अकारक-व्यत्यासः (Non-Karaka Divergence)

केचन शब्दाः भवन्ति, यैः सह अन्यस्य शब्दस्य विभक्तिः प्रभाविता भवति। ताः उपपदविभक्तयः भवन्ति। ताः षष्ठी शेषे (पा० .2.3.50) इत्यस्य अपवादस्वरूपाः सन्ति। एताः सर्वाः कारकेतरः विभक्तयः भवन्ति। एतस्मादस्य व्यत्यास्य 'अकारकव्यत्यासः' इति संज्ञा जाता।

विवरणम्

सहयुक्तेऽप्रधाने (तृतीया) (पा० 2-3-19)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

सहार्थेन युक्तेऽप्रधाने तृतीया विभक्तिः भवति।

संस्कृतम् सीता रामेण सह वनं गच्छति।

हिन्दी सीता राम के साथ वन जाती है।

उपर्युक्तोदाहणे संस्कृतभाषायां 'सीता' एवं 'वन' साक्षात् क्रियया सह अन्वेति। अतः तयोः कारकः सम्बन्धः वर्तते, तौ शब्दौ प्रधानौ स्तः। परन्तु रामः साक्षात् क्रियया सह नान्वेति, अतः तस्य कारकत्वन्नास्ति एवं च अप्रधानः शब्दः वर्तते। 'सह' शब्देन 'सीता एवञ्च राम' इत्येतयोः सम्बन्धः। तत्र अप्रधानः 'राम' वर्तते अतः तत्र तृतीयाविभक्तिर्भवति। परन्तु हिन्दीभाषायां 'राम' इति शब्दः 'साथ' इत्यनेन सह सम्बद्धः। तत्र सम्बन्धस्य भावः अतः तत्र षष्ठीविभक्तिर्भवति।

अग्रिमसरण्यां (सरणिः–5) एतादृशानां सूत्राणां सङ्कलनं कृतम्। यत्र संस्कृतभाषायाम् कारकेतरसम्बन्धाः सन्ति तदनुगुणं विभक्तिः भवति। तत्र हिन्दीभाषायां षष्ठी एवञ्च अन्यविभक्तयः भवन्ति।

क्रमाङ्कः	सूत्रसङ्ख्या	सूत्रम्	संस्कृतम्	हिन्दी
			विभक्तिः	विभक्तिः
1	1.4.84	तृतीयार्थे	2	6
2	1.4.85	हीने	2	6
3	1.4.87	उपोऽधिके च	2	6
4	1.4.89	लक्षणेत्थंभूताख्यानभागवीप्सासु प्रतिपर्यनवः	2	6
5	1.4.90	अभिरभागे	2	6
6	1.4.95	अतिरतिक्रमणे च	2	5,6
7	2.3.4	अन्तरान्तरेण युक्ते	2	6
8	2.3.5	कालाध्वनोरत्यन्तसंयोगे	2	7
9	2.3.10	पञ्चम्यपाङ्परिभिः	5	2
10	2.3.11	प्रतिप्रतिनिधिप्रतिदाने च यस्मात्	5	6
11	2.3.16	नमःस्वस्तिस्वाहास्वधाऽलंवषड्योगाच्च	4	2,4,6
12	2.3.19	सहयुक्तेऽप्रधाने	3	6
13	2.3.26	अन्यारादितरर्तेदिक्छब्दाञ्चूत्तरपदाजाहियुक्ते	5	5,6

सरणिः – 5 अकारक–व्यत्यासः

क्रियास्वरूप–व्यत्यासः (Verbal Divergence)

एषः क्रियामाधारीकृत्य व्यत्यासः भवति। संस्कृतभाषायां काश्चन क्रियाः एतादृशाः भवन्ति येषां यत् कर्म

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

भवति तद् कर्म हिन्दी भाषायां आधारः भवति। तस्य कारणं हिन्दीभाषायां ताः क्रियाः अकर्मकत्वेन बोधिताः सन्ति। अतः हिन्दीभाषायां तस्य आधारत्वेन विवक्षितत्वात् तत्र सप्तमीविभक्तिर्भवति। वस्तुतः संस्कृतभाषायां एतादृशः क्रियाबोधकः यः धातुर्वर्तते सः सकर्मकः। परन्तु हिन्दीभाषायां तस्य स्वरूपं परिवर्तितम्। तत्र सः धातुः अकर्मकः जातः। अतः क्रियास्वरूपभेदात् अयं व्यत्यासः 'क्रियास्वरूपव्यत्यासः' कथितः।

विवरणम्

संस्कृतम् वानरः वृक्षम् आरोहति।

हिन्दी बन्दर वृक्ष पर चढता है।

उपर्युक्तोदारहणे 'आ+रुह' अयं धातु संस्कृतभाषायां सकर्मकः अत एवात्र 'वृक्ष' इत्यस्य कर्मसंज्ञा भवति तस्य फलं तत्र द्वितीयाविभक्तिः भवति। परन्तु हिन्दीभाषायां 'चढ' इति धातुः अकर्मकः। अतः 'वृक्ष' इत्यस्य आधारत्वेन विवक्षितत्वात् तत्र सप्तमीविभक्तिः भवति। अनेन प्रकारेण ये धातवः संस्कृतभाषायां सकर्मकाः परन्तु हिन्दीभाषां अकर्मकाः तत्र एतादृशः व्यत्यासः दृश्यते।

संस्कृत-हिन्दीभाषयोः क्रियां स्वीकृत्य यदि कर्मकारकस्य भेदाः वर्तन्ते तेषां वर्गीकरणं यान्त्रिकानुवादं साहाय्यं विधास्यति।

क्रियामूल-व्यत्यासः (Complex-Predicate Divergence)

संस्कृतभाषायां यत्र क्रिया एका भवति परन्तु हिन्दीभाषायां तस्याः परिवर्तनं क्रियामूलरूपेण भवति। क्रियामूलरूपेण परिवर्तनेन अयं व्यत्यासः 'क्रियामूलव्यत्यासः' जातः।

विवरणम्

संस्कृतम् सीता रामम् अनुसरति।

हिन्दी सीता राम का अनुसरण करती है।

उपर्युक्तोदाहरणे 'अनु+सृ' इति धातोः कर्म 'राम' अस्ति अतः तत्र द्वितीया विभक्तिः। परन्तु हिन्दीभाषायां 'अनुसंरण कर' इति धातुः क्रियामूलः धातुः वर्तते अतः क्रियामूलधातुयोगे कर्मणि षष्ठीविभक्तिः भवति। अत्र वस्तुतः 'अनुसरण' इति पदं संस्कृताधारितं कृदन्तं वर्तते। अतः 'कर्तृकर्मणोः कृतिः' (पा० 2.3.65) इत्यनेन षष्ठीविभक्तिः भवति। अयं नियमः हिन्दीभाषायामपि प्रवर्तते।

क्रियामाधारीकृत्य व्यत्यासां विशदरूपेण अध्ययनमपेक्ष्यते। तत्र द्विसहस्रधातुनां कारकाकाङ्क्षा एवं हिन्दीभाषया सह तुलनात्मकमनुशीलमपेक्षते। एतादृशमध्ययनं यान्त्रिकानुवादे अत्यावश्यकं महत्त्वपूर्णञ्च वर्तते।

उपसंहारः

व्यत्यासप्रकाराणामिदानीमेतेन प्रकारेण उपसंहृयते –

(अ) व्यत्यासः संस्कृतव्याकरणस्य प्रभावेन अग्रिमसरण्यां स्पष्टीक्रियते

क्रमाङ्कः	व्यत्यास-प्रकारः	संस्कृतम्	हिन्दी
1	वैकल्पिक-व्यत्यासः	वैकल्पिकरूपेण विभक्ती द्वे प्राप्येते।	उत्सर्गतः एकैव विभक्तिः भवति।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

2	आपवादिक–व्यत्यासः	उत्सर्गविभक्तिं प्रबाध्य इतरविभक्तिनिर्धारणं कुर्वन्ति।	उत्सर्गनियमेन एव विभक्तिनिर्धारणं भवति।
3	भेदज–व्यत्यासः	भिन्ना विभक्तिः भवति	भिन्ना विभक्तिः भवति।
4	अन्यतर–व्यत्यासः	संस्कृतभाषायां बहुविभक्तयः प्राप्ताः।	एकामेव विभक्तिं स्वीकरोति।
5	अकारक–व्यत्यासः	अकारकेषु एकाधिकविभक्तयः भवन्ति।	प्रायः षष्ठीविभक्तिरेव भवति।
सरणिः – 6 उपसंहारः			

(ब) हिन्दीभाषायाः विशिष्टस्वरूपत्वेन व्यत्यासः अधो लिखितम् वर्तते।

1. क्रियायाः विशेषविभक्तिनिर्धारणम्।
2. क्रियामूलम्

उपर्युक्तविवेचनाधारेण यान्त्रिकानुवादे शब्दानां अर्थनिर्धारणे एका पद्धतिः निर्मिता भवति। एवञ्च संस्कृतभाषायां एवमाधुनिकभारतीयभाषासु विभक्तिस्तरे व्यत्यासानां विवरणं अनया रीत्या प्रतिपादयितुं शक्यते।

कृतज्ञता –ज्ञापनम्

कार्यमिदं भारतसर्वकारस्य "संस्कृत–सङ्गणंक–साधनानि एवं संस्कृत–हिन्दी–यान्त्रिकानुवाद–यन्त्रस्य निर्माणम्" (Development of Sanskrit Computational tools and Sanskrit–Hindi Machine Translation System) इति नाम्नी परियोजनायाः भागं वर्तते। सम्मान्यान् प्रो० रामकृष्णमाचार्यमहाभागान् एवं प्रो० लक्ष्मीबाईमहोदयाः सविनयं कार्तज्ञं विज्ञापयामः। येषां मार्गदर्शनेन कार्योऽयं समाप्तिमगात्।

सन्दर्भग्रन्थसूची

1. आचार्य रामशास्त्री, 'संस्कृत शिक्षण सरणी', आचार्य रामशास्त्रि ज्ञानपीठ (१९९८)
2. द्विवेदी, कपिल देव: 'भाषा विज्ञान एवं भाषा शास्त्र', विश्वविद्यालय प्रकाशन, वारणसी (२००६)
3. शर्म, राम रंग तथा शर्मा, मालती: 'श्री भट्टोजिदीक्षित विरचित वैयाकरण सिद्धान्तकौमुदी' (कारक प्रकरणम्), भारतीय विद्या प्रकाशन, वारणसी–देहली (१९९७)
4. Cardona, George: 'Panini and Paniniyas on sesa relations', Kunjunni Raja Academy of Indological Research, Kochi (2007)
5. Dorr, Bonnie: 'Classification of Machine Translation Divergence and a Proposed Solution', Computational Linguistics 20(4) (1994)
6. Dave, S., Parikh, J., Bhattacharya, P.: 'Interlingua Based English–Hindi Machine Translation and Language Divergence', Journal of Machine Translation (JMT) 17 (2002)
7. Goyal, Pawan and Sinha, R. Mahesh K.: 'Translation Divergence in

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

English–Sanskrit–Hindi Language Pairs', In Amba Kulkarni, and Gerard Huet, editors, Sanskrit Computational Linguistics, Proceedings of Third International Symposium, Hyderabad. Springer–Verlag LNAI 5406 (2009)

8. Mishra, Vimal and Mishra, R. B.: 'Study of Example Based English to Sanskrit Machine Translation', Journal of Research and Development in Comp Sc. And Engg Number 37, Jan–June (2008)

9. Sharma, Rama Nath: 'The Astadhyayi of Panini', Vols 1–3, Munshiram Manoharlal, Delhi 2nd Ed (2002)

10. Vasu, Srisa Chandra: 'The Siddhanta Kaumudi of Bhattoji Dikshita', Vol I, Motilal Banarsidass, Delhi (2003)

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

संस्कृतसङ्गणकसाधनसमुच्चयस्य संस्कृत–हिन्दीयन्त्रानुवादतन्त्रांशस्य व्यवस्थायाश्च निर्माणम्

पृष्ठभूमिका:

भाषाप्रक्रमः (Natural Language Processing) भाषाप्रक्रमः गणकीयप्रतिरूपनिर्माणं लक्ष्यीकृत्य प्रवर्तते। गणकीयप्रतिरूपः भाषायाः विश्लेषणाय प्रजननाय च उपकरोति। अस्य तान्त्रिकप्रयोजनन्तु यन्त्रानुवादव्यवस्था, भाषामाध्यमात् दत्तनिधिः, सङ्गणकयन्त्रस्य **मानवीय–यन्त्रीयमाध्यमं**, भाषावबोधव्यवस्था, व्यवस्थावगमनम्, पाठविश्लेषणम्, गणकीयोपकृतशिक्षाव्यवस्था, मुद्रितपाठं आहोस्वित् हस्तलिखितपाठं पठितुम् अवगन्तुञ्च समर्थं गणकयन्त्रम् इत्यादीनाम् उत्पादनम्। भाषाप्रक्रमे सामग्री अस्ति– व्याकरणनियमाः, विधिकल्पः, दत्तांशरचना, प्रापञ्चिकपरिज्ञानप्रदर्शकविधिः, तर्कयुक्तिश्च। इयं सामग्री सङ्गणकपरिज्ञानं, कृतकधीः, भाषाविज्ञानं, तर्कः, तत्त्वज्ञानं इत्यादिभ्यः सङ्कलिता।

संस्कृतगणकीयभाषाविज्ञानम् –अन्तर्विद्याविषयकक्षेत्रम् (Sanskrit Computational Linguistics – An interdisciplinary field)

यन्त्रानुवादव्यवस्थाकल्पनाय गणकीयभाषाविज्ञानं प्रादुर्भूतम्। 85 विश्वविद्यालयेषु गणकीयभाषाविज्ञानम् अधीयते। रङ्गेऽस्मिन् नैके शोधगणाः अहरहरं परिश्रमं कुर्वन्ति। भाषाविज्ञानस्य अभिनवोऽऽविष्कारः अस्ति संस्कृतसङ्गणकभाषाविज्ञानमिति। संस्कृतसङ्गणकभाषाविज्ञानक्षेत्रे आ दशकात् नैके सङ्गणकभाषाविज्ञानिनः पारम्परिकसंस्कृतव्याकरणशास्त्राध्येतारश्च शोधरताः सन्ति। तस्मात् संस्कृतगणकीयभाषाविज्ञानम् अन्तर्विद्याविषयकक्षेत्रत्वेन ख्यातिं गतोऽस्ति। क्षेत्रमिदम् अधिकतया पाणिनीयव्याकरणसम्प्रदायमनुसृत्य प्रवर्तते। यतो हि पाणिनिना तदानीन्तन व्यावहारिकसंस्कृतभाषायाः विश्लेषणाय प्रवृत्तेन नैकानि भाषाशास्त्रस्य मूलतत्त्वानि आविष्कृतानि यानि च अन्यभाषाविश्लेषणेऽपि शक्यसमन्वयानि। अत एव भाषाविज्ञानक्षेत्रे प्रचुरतया पाणिनीयसिद्धान्ताः उपयुक्ताः। अन्यच्च संस्कृतसाहित्यं विज्ञानभाण्डागारोऽस्ति।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

<u>यन्त्रानुवादव्यवस्था</u>– अशीतितमवर्षादारभ्य यन्त्रानुवादव्यवस्थायाः विषये महती आसक्तिः लोके परिलक्ष्यते। भूयान् परिश्रमोऽपि विधीयते। यूरोपीयभाषाणां कृते 'युरेत्रा' ' Eurotra', जपानभाषायाः आङ्ग्लभाषायाश्च कृते 'म्यू' 'Mu', Carnegie Mellon विश्वविद्यालये आङ्लजपान्भाषयोर्कृते ' KBMT' भारतीयभाषाणां कृते कान्पुरे 'अनुसारकम्' अन्यत्र आङ्गलभारती, शक्तिः, मात्रा, मन्त्रश्चेति यन्त्रानुवादव्यवस्थाः' जगत्यस्मिन् दरीदृश्यन्ते।

<u>संस्कृतसङ्गणकयन्त्रानुवादः –</u> संस्कृतक्षेत्रे यन्त्रानुवादः प्रयोगदशायामेव वर्तते। प्राचार्यः ह्यूयट् महाभागः इन्ना फ्रान्स इत्यत्र फ्रेन्चपाठकानां सहायार्थं संस्कृतग्रन्थपठनाय कानिचन पठनसाधनानि प्रकल्पयति। प्राचार्यः पीटर् ब्रौन् विश्वविद्यालये, ष्राफ तद्गणश्चापि आङ्ग्लपाठकानां कृते च पठनपरिकरम् समुपकल्पयति। भारतेऽपि ए.एस्सार्. मेलकोटे, सि.डाक् बङ्गलूरु, जवहरलालनेहरुविश्वविद्यालयश्च सङ्गणकयन्त्रानुवादपरिकरनिर्माणे कृतभूरिपरिश्रमः वर्तते।

यन्त्रीयविधानेन वा मानवीयेन यत्नेन वा मानवीयेन यत्नेन विनैव वा गणकीयव्यंवस्था अनुवादं सम्पादयति चेत् यन्त्रानुवादः इति कथ्यते। यन्त्रानुवादस्तु अन्तर्विद्याविषयकक्षेत्रं यतो हि नैकानां विषयाणां समाहारः अत्र परिलक्ष्यते। कृतकधीः, भाषाविज्ञानं, सङ्गणकपरिज्ञानं, अनुवादसिद्धान्ताः, नैकानि तन्त्राणि च सम्भूय प्रवर्तन्ते।

<u>उपलभ्यमानानि साधनानि–</u> संस्कृत–हिन्तीयन्त्रानुवाद इति परियोजनाया अस्या आरम्भात्पूर्वमेव भागग्रहीतृभिः काभिश्चत् संस्थाभिः कानिचन यन्त्रानुवादसाधनानि प्रकल्पितानि। तानि च–

1. Morphological analyser engine शब्दरूपविश्लेषकयन्त्रम्, Hindi generator हिन्दीभाषाप्रजनकम्, POS Tagger वाक्यांशपदवर्गाङ्कनञ्चेति सङ्गणकघटकाः ILMT consortium इत्यनेन प्रकल्पिताः।
2. Morphological analyser शब्दरूपविश्लेषकयन्त्रम्, sandhi splitter सन्धिविच्छेदकम् इत्याद्यः सामग्र्यः तिरुपतिस्थराष्ट्रीयसंस्कृतविद्यापीठस्य साहाय्येन हैदराबादस्थकेन्द्रीयविश्वविद्यालयेन विनिर्मिताः।
3. Morphological analyser engine शब्दरूपविश्लेषकयन्त्रम्, POS Tagger वाक्यांशपदवर्गाङ्कनम्, अमरकोशः, e–संस्कृतपाठसंग्रहश्चेति देहलीस्थजवहरलालनेहरुविश्वविद्यालयेन उपकल्पितम्।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

4. e- संस्कृतपाठसंग्रहश्चेति SanskNet संस्कृतजालः इति परियोजनायाः अन्तर्गततया तिरुपतिस्थेन राष्ट्रीयसंस्कृतविद्यापीठेन उत्पादितम्।

इमाः संस्थाः स्वप्रकल्पितघटकान्, सामग्रीं, साधनानि, e-संस्कृतपाठसंग्रहञ्च समादाय, सम्भूय च अन्याः काश्चित् संस्था अपि समायोज्य **संस्कृत-हिन्दी यन्त्रानुवादकस्य** निर्माणे निरताः वर्तन्ते।

<u>संस्कृतसङ्गणकसाधनसमुच्चयस्य संस्कृत-हिन्दीयन्त्रानुवादतन्त्रांशस्य व्यवस्थायाश्च निर्माणम्- इति परियोजनायां कार्यरतं संस्थासप्तकम्-</u>

परियोजनाया अस्याः धुरं वहति हैदराबादनगरस्थकेन्द्रीयविश्वविद्यालयस्य संस्कृतविभागस्य अध्यक्षा प्राचार्या विदुषी अम्बाकुलकर्णिवर्या। प्रस्तूयतेऽत्र संस्थासप्तकस्य विवरणम्-

क्र.सं.	संस्था	सदस्यः
1.	हैदराबादनगरस्थकेन्द्रीयविश्वविद्यालयः हैदराबाद् आन्ध्रप्रदेशः	प्राचार्या अम्बाकुलकर्णी- अध्यक्षा संस्कृताध्ययनकेन्द्रम् प्रधानसंयोजिका
2.	जगद्गुरुरामानन्दाचार्यराजस्थानसंस्कृतविश्वविद्यालयः, जयपुरम् राजस्थानम्	प्राचार्यः वीरनारायण पाण्डुरङ्गिवर्यः
3.	जवहरलालनेहरूविश्वविद्यालयः देहली	प्राचार्यः गिरीशनाथझा
4.	संस्कृत अकादमी हैदराबाद आन्ध्रप्रदेशः	आचार्यः श्रीनिवास वरखेडी निदेशकः
5.	राष्ट्रियसंस्कृतविद्यापीठम्, तिरुपतिः आन्ध्रप्रदेशः	प्राचार्यः एस्. एस्. मूर्तिः
6.	पूर्णप्रज्ञाविद्यापीठम्, बङ्गलूरु	आचार्यः तिरुमलकुलकर्णी
7.	अन्तराष्ट्रियसमाचारसाङ्केतिकसंस्था, हैदराबाद् आन्ध्रप्रदेशः	आचार्या दीप्तिशर्ममिश्रवर्या

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

प्रधानं लक्ष्यम्– संस्कृत–हिन्दीयन्त्रानुवादव्यवस्थायाः परिकल्पना संस्कृतसङ्गणकसाधनानाम् उत्पत्तिश्च।

प्रशासनीयानुमतिः 2008 तमे वर्षे मार्चमासस्य 28 तमे दिनाङ्के देहलीस्थसञ्चारसूचनाप्रौद्योगिकी मन्त्रालयस्य सूचनाप्रौद्योगिकीविभागेन प्रदत्ता प्रशासनीयानुमतिः।

कालावधिः– वर्षत्रयम् अथवा (36)षट्त्रिंशत्मासात्मकः कालावधिः । 2008 तमवर्षस्य एप्रिल् मासे समारब्धेयं परियोजना।

कार्यविभजनम्– यन्त्रानुवादकनिर्माणाय शतशः वर्षाणि अपेक्षितानि। तथापि वर्षत्रये महल्लक्ष्यमिदं साधनीयमिति कार्यविभजनं कृतम् गणशः कालशश्च ।

लक्ष्यसाधनकालावधिः–कालशः कार्यविभजनम्

मासः

आरम्भः–समाप्तिः	लक्ष्यम्
1 – 3	Corpora collection e–संस्कृतपाठसंग्रहः
1 – 3	Evaluation of existing tools उपलभ्यमानायाः सामग्र्याः मूल्याङ्कनम्
1 – 3	Documentation on standards मानकस्वरूपसंरचना
1 – 6	Morph analyser Engine शब्दरूपविश्लेषकयन्त्रम्
1 – 6	POS tagging Engine वाक्यांशपदवर्गाङ्कनयन्त्रम्
4 – 9	Documentation on standards मानकस्वरूपसंरचना
4 – 30	Corpora Annotation साहित्यसङ्ग्रहस्य विश्लेषणम्
4 – 30	Bilingual dictionaries द्विभाषाकोशाः

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri

4	–	30	Morph analyser data शब्दरूपविश्लेषकदत्तांशः
4	–	12	Sandhi engine सन्धिविच्छेदकयन्त्रम्
4	–	15	Samaasa Tagger समाससूचनाबन्धग्रथनम्
4	–	24	Morph Analyser शब्दरूपविश्लेषकम्
12	–	24	Lexical disambiguation engine पदसूचीनिर्णायकयन्त्रम्
12	–	18	Evaluation procedure परीक्षणसंविधानम्
13	–	36	Testing and integration परीक्षणं संयोजनम् अनुसन्धानम्
15	–	24	Generator engine प्रजननयन्त्रम्
18	–	30	Transfer Grammar स्थानान्तरीकरणव्याकरणम्
22	–	28	Transfer Engine स्थानान्तरीकरणयन्त्रम्
22	–	33	Full fledged parser वाक्यस्य पूर्णविश्लेषकम्
34	–	36	Delivery of the product उत्पादनस्य प्रदानम्

गणशः कार्यविभजनम्

संस्था	लक्ष्याणि
1. केन्द्रीयविश्वविद्यालयः –	1. सन्धिसमासयोः व्याकरण transfer lexical disambiguation सम्बद्धयन्त्रनिर्माणम् 2. सन्धिसमासकारकाणां कृते गणकीयसूचीबन्धग्रथितदत्तांशस्य निर्माणम्

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

3. Develop Lexical disambiguation rules कोशीयपदसूचीनियमानां संरचना
4. Develop transfer grammar rules/data व्याकरणनियमसंरचना
5. Testing integration and overall management परीक्षणं, संयोजनं सर्वविधप्रबन्धनम्
6. Develop standards for different levels of tagging विभिन्नस्तरीयसूचनाबन्धग्रथनाय मानकस्वरूपनिर्धारणम्

2. जवहरलालनेहरूविश्वविद्यालयः

1. नियमाधारितवाक्यांशपदवर्गाङ्कनयन्त्रम्
2. वाक्यांशपदवर्गाङ्कनाय, सन्धिसमासकारकेभ्यः सूचीबन्धग्रथितदत्तांशस्य निर्माणम्
3. द्विभाषाकोशनिर्माणम्
4. Develop lexical disambiguation rules

उ. develop transfer grammar rules/data

ऊ.नैकस्तरीयसूचनाबन्धग्रथनाय

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri

	मानकस्वरूपनिर्धारणम्
3. Jagadguru Ramanandacharya Rajasthan Sanskrit University	1. वाक्यांशपदवर्गाङ्कनाय, सन्धिसमासकारकेभ्यः सूचीबन्धग्रथितदत्तांशस्य निर्माणम् 2. द्विभाषाकोशनिर्माणम् 5. Develop lexical disambiguation rules
4. राष्ट्रीयसंस्कृतविद्यापीठम्, तिरुपतिः	1. वाक्यांशपदवर्गाङ्कनाय, सान्धिसमासकारकेभ्यः सूचीबन्धग्रथितदत्तांशस्य निर्माणम्
5. पूर्णप्रज्ञविद्यापीठम्, बङ्गलूरु	1. वाक्यांशपदवर्गाङ्कनाय, सन्धिसमासकारकेभ्यः सूचीबन्धग्रथितदत्तांशस्य निर्माणम्
6. संस्कृत अकाडमी, हैदराबाद्	1. उपलभ्यमानानां साधनसामग्याः मूल्याङ्कनम् 2. यन्त्रानुवादव्यवस्थायाः उत्पादनावसरे विभिन्नस्तरेषु परीक्षणं मूल्याङ्कनम् 3. वाक्यांशपदवर्गाङ्कनाय, सन्धिसमासकारकेभ्यः सूचीबन्धग्रथितदत्तांशस्य निर्माणम्
7. अन्तराष्ट्रियसमाचारसाङ्केतिकसंस्था, हैदराबाद् आन्ध्रप्रदेशः	1. विभिन्नस्तरीयसूचीबन्धग्रथनाय मानकस्वरूपोत्पत्तिः

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

2. परीक्षणं मूल्याङ्कनम्

3. संस्कृतवाक्यविश्लेषकम्

संस्कृत -हिन्दीयन्त्रानुवादः संस्कृत-हिन्दीभाषयोर्मध्ये बह्वंशेषु साम्यमस्ति। संस्कृतभाषायाः रूपान्तरीकरणेनैव प्रादुर्भूता वर्तते हिन्दीभाषा। द्वयोर्भाषयोः शब्दजालः, वाक्यनिर्माणञ्च साम्यं भजतः। संस्कृतभाषां हिन्दीभाषायाम् अनुवचनं कर्तुं शक्यसाध्यम्।

अनुवचनकर्मणि अनुसृतः सोपानक्रमः-

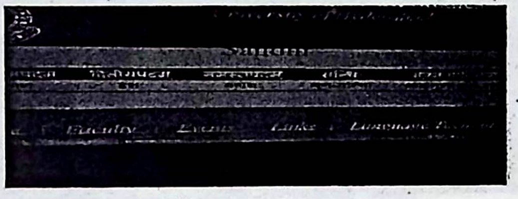
e-संस्कृतसाहित्यसङ्ग्रहः (Corpora collection)- नवदशशतके संस्कृतसाहित्यजालपरियोजना Sanskrit Net Project नाम्नी काचित् परियोजना तिरुपतिस्थेन राष्ट्रियसंस्कृतविद्यापीठेन स्वीकृता यस्याञ्च सहस्राधिकग्रन्थानां e-सङ्ग्रहः कृतः। e-सङ्ग्रहोऽयं संस्कृत-हिन्दीयन्त्रानुवादाय उपयुज्यते। विभिन्नस्तरेषु e-सङ्ग्रहः सूचीबन्धैः ग्रथ्यते चेत् यन्त्रानुवादव्यवस्थानिर्माणं सेत्स्यति।

e-संस्कृतसाहित्यसङ्ग्रहस्य सूचीबन्धग्रथनम् - सन्धिविच्छेदनाय, वाक्याङ्कपदवर्गाङ्कनाय, समासाङ्कनाय, कारकाङ्कनाय, वाक्यविश्लेषणाय च सूचीबन्धाः नितराम् उपकारकाः भवन्ति। प्रत्येकस्मिन् स्तरे संस्कृतवैयाकरणैः, संस्कृतभाषाविज्ञानिभिः सङ्गणकज्ञैश्च सूचीबन्धाः निश्चीयन्ते। e-संस्कृतसाहित्यसङ्ग्रहे आरम्भे तावत् मानवीयेन यत्नेन सूचीबन्धाः अङ्क्यन्ते।

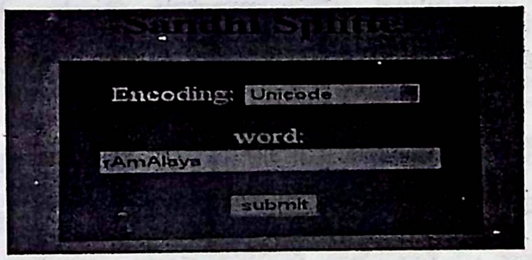
सन्धिविच्छेदनम् - सन्धिविच्छेदकम् (Sandhi Splitter) e-संस्कृतपाठस्य सन्धिविच्छेदनं नियमानुगुणं मानवीयेन यत्नेन क्रियते। एतदाधारेण सन्धिविच्छेदकः कश्चित् यन्त्रप्रक्रमः उत्पद्यते। यन्त्रप्रक्रमेणऽनेन अपेक्षितस्थलेषु सन्धिविच्छेदनं सम्पद्यते। सन्धिविच्छेदनपूर्वकपाठ्यांशः अग्रिमविश्लेषणाय सज्जीक्रियते।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.



सन्धिविच्छेदकेन कृतम् सन्धिविच्छेदनम्



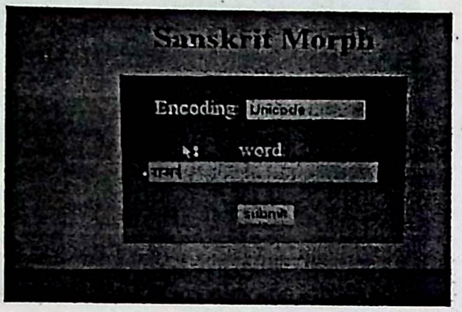
सन्धिविच्छेदकम् – Sandhi Splitter

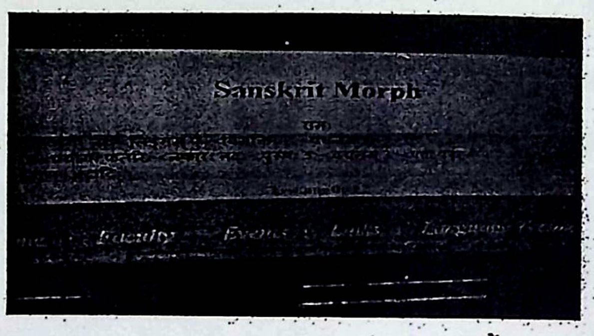
संस्कृतभाषाविश्लेषणं –संस्कृतभाषाविश्लेषकयन्त्रम् –(Sanskrit Morphological Analyzer) विश्लेषणप्रक्रियायां सोपानद्वयं वर्तते। प्रथमे सोपाने तु सन्धिविश्लेषणं, सुबन्तविश्लेषणं, तिङ्न्तविश्लेषणं, वाक्यांशपदवर्गनिर्धारणं समासविश्लेषणञ्चेति। द्वितीये सोपाने कारकविश्लेषणं क्रियते। एषु स्तरेषु प्रथमं नियमावल्यनुगुणं मानवीयेन यत्नेन सूचीबन्धाः अङ्क्यन्ते।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

संस्कृतभाषाविश्लेषकयन्त्रे बहूनि अङ्गानि निर्मितानि। सन्धिविच्छेदकं, सुबन्तविश्लेषकं लिङ्गानुशासकं तिङ्गन्तविश्लेषकं समासविश्लेषकं वाक्यांशपदवर्गनिर्धारकं कारकविश्लेषकं चेत्यादयो भागाः सन्ति। सर्वेऽपीमे भागाः पृथक् निर्मीयन्ते। अनन्तरं संयोज्यन्ते च।

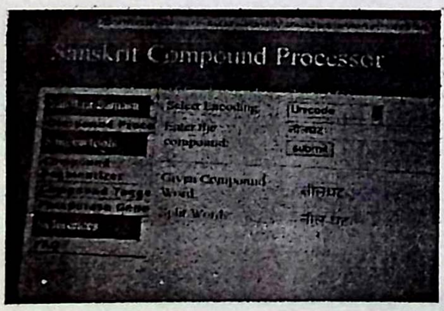




संस्कृतभाषाविश्लेषकम्

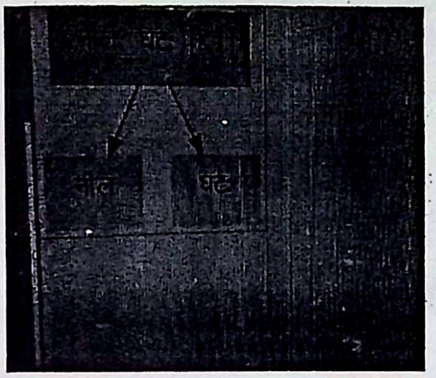
संस्कृतवाक्यांशपदवर्गाङ्गनम् – सन्धिविच्छेदिते संस्कृतपाठांशे वाक्यांशपदवर्गाङ्गनं क्रियते। नो चेत् अन्वयनिरूपणं दुश्शकं भवति। अपि च यन्त्रानुवादप्रक्रियायां सन्दिग्धता दूरीक्रियते। वाक्यविश्लेषणञ्च सम्पद्यते।

समासानां सूचनाबन्धग्रथनम्–समासविश्लेषकम्– संस्कृतसाहित्ये आदौ मानवीयेन यत्नेन समस्तपदानि सूचनाबन्धैः ग्रथ्यन्ते। तदाधारेण यन्त्राध्ययनाभियुक्त्या यन्त्रीयसमस्तपदाभिज्ञाता Samaasa Identifier उत्पद्यते। अनेन विग्रहवाक्यं समासभेदश्च प्रदर्श्यते।



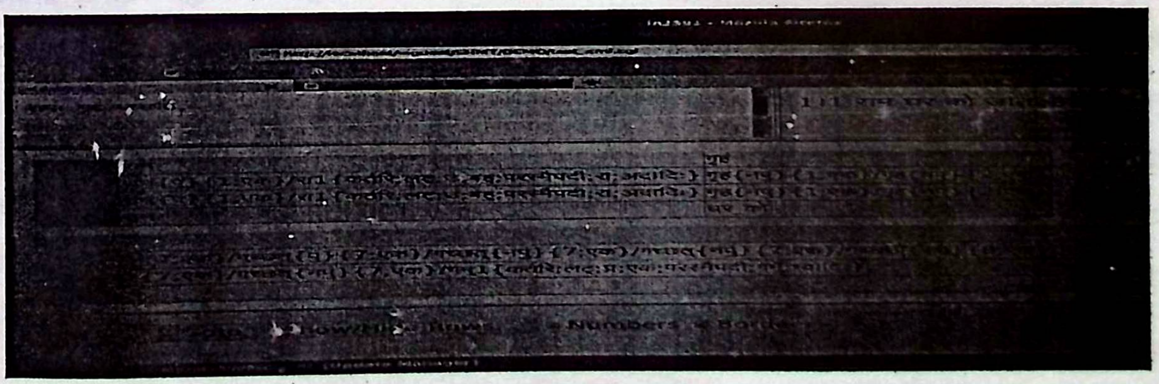
CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.



कारकीयस्तरे सूचनाबन्धग्रथनम्– स्तरेऽस्मिन्नपि मानवीयेन यत्नेन सूचनाबन्धाः अङ्क्यन्ते। Kaaraka Tagger कारकाङ्कनिर्धारकं तदनु यन्त्रीयविधानेन कारकविश्लेषणं करोति। तत्पश्चात् यन्त्रीयवाक्यविश्लेषकम् उत्पाद्यते।

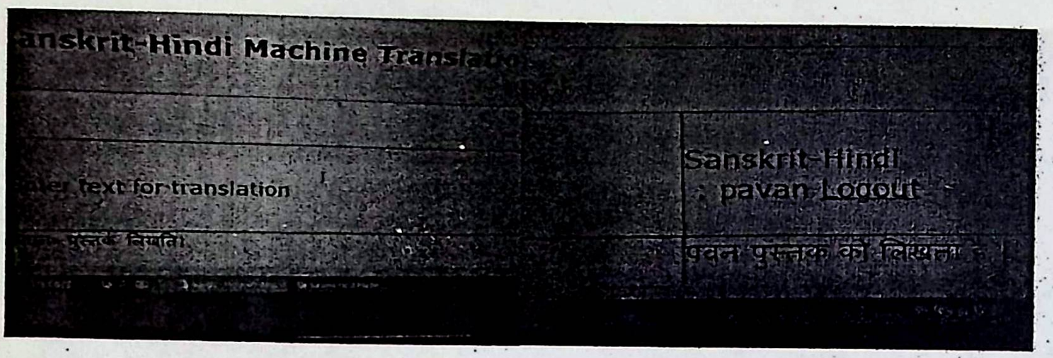
कार्यप्रगतिः 480 किलोऽङ्कमितं सूचीबन्धग्रथितं e–संस्कृतसाहित्यम् अधुना यन्त्रानुवादाय उपलभ्यते। शब्दरूपविश्लेषकं, वाक्यांशपदवर्गाङ्कनयन्त्रञ्च अनुसन्धानितौ समायोजितौ उपलभ्येते च। Transfer engine Hindi Generator हिन्दीभाषाप्रजनकं च ILMT consortium भारतीयभाषायन्त्रानुवादपरियोजनातः समाद्रियेते। सन्धिविच्छेदकं, samaasa tagger समासाङ्कनं कार्यशीलौ, अनुसन्धानितौ संयोजितौ च। वाक्यविश्लेषककार्यं सम्पन्नम्। द्विभाषाकोशः समुपलभ्यते। आप्टे कोशस्य e–प्रारूपं विनिर्मितम्। विभिन्नस्तरेषु सूचीबन्धग्रथनाय मानकस्वरूपः निर्धारितः। निर्धारितसूचीबन्धग्रथननियमावलिः ग्रन्थस्थीकृता च। संस्कृत–हिन्दीभाषानुवादसाधनसामग्र्याः परीक्षणं मूल्याङ्कनञ्च सम्पन्नम्। यन्त्रानुवादकस्य सर्वाणि अङ्गानि विनिर्मितानि। पार्थक्येन स्वकार्यं निर्वहन्ति च। सद्यः तेषां संयोजनमपि सम्पन्नम्।



CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

यन्त्रानुवादः



प्रगतिसमीक्षात्मकगोष्ठ्यः-ज्ञानसत्राणि-कार्यशालाः

यन्त्रानुवादप्रक्रियायाः परिचयः, परस्परज्ञानविनिमयः, प्रशिक्षणं, काले काले समीक्षणञ्च परियोजनायाः साफल्याय कारणीभूताः भवन्ति।

संस्कृत -हिन्दीयन्त्रानुवादपरियोजनायाः प्रगतिसमीक्षात्मकगोष्ठ्यः-PROGRESS REVIEW STEERING GROUP MEETING: यन्त्रानुवादे अनुभवज्ञाः भाषाविज्ञानिनः सङ्गणकज्ञाः प्रगतिसमीक्षात्मकसमितेः सदस्याः भवन्ति। समितिसदस्याः षण्मासेषु एकवारं गोष्ठीं रचयित्वा, परियोजनासदस्यानां परियोजनानिवेदिकां श्रुत्वा, प्रगतिसमीक्षणं कृत्वा, गुणदोषान् विचार्य, आवश्यकसूचनाः यच्छन्ति। अस्याः समितेः सदस्याः सन्ति प्राचार्यः राजीवसङ्गाल्, निदेशकः, अन्तराष्ट्रियसमाचारसाङ्केतिकसंस्था, हैदराबाद्, आन्ध्रप्रदेशः, प्राचार्यः लक्ष्मीश्वरझा, SLBSRSV, New Delhi, श्रीमती सुनीता अरोरा, CDAC, NOIDA, श्रीकीर्तिकान्तशर्मा, IGNCA, New Delhi, प्राचार्यः आर्.सि.पाण्डा. बनारस् हिन्दूविश्वविद्यालयः, वाराणसी, श्रीमती स्वर्णतला, निदेशिका, विभागाध्यक्षा, (HCC) DIV, DIT, Delhi, श्रीविजयकुमारः, वैज्ञानिकः, 'E' TDIL, DIT, Delhi च।

ज्ञानसत्राणि- यन्त्रानुवादप्रक्रिया तावत् क्लिष्टा वर्तते। नैके स्तराः सन्ति। स्तरानुगुणं नियमाः भवन्ति। यन्त्रानुवादप्रक्रियायाः, व्याकरणशास्त्रस्य, भाषाविज्ञानशास्त्रस्य च आधारेण नियमावलिः रचनीया। तत्तत्क्षेत्रज्ञाः अनुभवज्ञाः ज्ञानसत्रेषु यन्त्रानुवादानुगुणां सूचनाबन्धग्रथननियमावलिं रचयित्वा साध्यासाध्यान् विचारयन्ति।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

समासकारकयोः सूचनाबन्धग्रथननियमावलिः वैयाकरणैः प्राचार्यैः रामकृष्णमाचार्यैः रचिता। वाक्यांशपदवर्गाङ्कननियमावलिस्तु जवहरलालनेहरूविश्वविद्यालयीयसंस्कृताध्ययनविभागस्य हैदराबादनगरस्थायाः अन्तराष्ट्रियसमाचारसाङ्केतिकसंस्थायाः च भारतीयभाषायन्त्रानुवादव्यवस्थायाः नियमावलिम् आधारीकृत्य आचार्यैः श्रीनिवासवरखेडीवर्यैः रचिता। हैदराबादनगरस्थकेन्द्रीयविश्वविद्यालये हैदराबादनगरस्थसंस्कृत अकाडमीसंस्थायां च दिनद्वयात्मकं वाक्यांशपदवर्गाङ्कनसम्बद्धं ज्ञानसत्रं संयोजितम्।

कार्यशालाः– संस्कृतभाषायन्त्रानुवादक्षेत्रं तावत् अन्तर्विद्याक्षेत्रमिति उक्तपूर्वमेव। नैकाः संस्थाः, संस्कृतज्ञाः विशिष्य वैयाकरणाः, गणकज्ञाः, संस्कृतभाषाविज्ञानिनः, शोधगणाः अन्ये च परियोजनायामस्यां तल्लीनाः वर्तन्ते। ऐदम्प्राथम्येन संस्कृत–हिन्दीभाषयोः यन्त्रानुवादः समारब्धः इत्यतः शोधगणाः अनुभवराहित्येन कार्यं कुर्वन्ति। अतः स्तरानुगुणं विभिन्नसोपानेषु शोधगणस्य प्रशिक्षणाय कार्यशालाः नैकस्थलेषु समायोज्यन्ते। नवत्यधिकविंशतितमे वर्षे मई मासे बङ्गलूरुनगरस्थे पूर्णप्रज्ञाविद्यापीठे एकसप्ताहिककार्यशाला समासाङ्कनाय समायोजिता। सप्तसंस्थाभ्यः परियोजनायाः सदस्याः शोधगणाश्च भागम् अगृह्णन्। प्रशिक्षिताः शोधगणाः स्वसंस्थाः प्रतिगत्य समासाङ्कनम् अकुर्वन्। संस्कृत अकाडमीसंस्थायां त्रिदिवसीया कारकाङ्कनसम्बद्धा कार्यशाला आयोजिता यस्याञ्च शोधगणाः कारकाङ्कनविषये अवगाहनं प्राप्नुवन्। अस्मिन् विषये एव पूनानगरे CASS, पूनाविश्वविद्यालये डिसम्बर् मासे कार्यशाला आयोजयिष्यते।

संस्कृत –हिन्दीभाषायन्त्रानुवादसम्बद्धसाहित्यसंरचना– यन्त्रानुवादप्रक्रियायां नैकान् विषयान् समालोढ्य ये नियमाः रचिताः, या प्रक्रिया च आविष्कृता तत्सर्वं ग्रन्थस्थीक्रियते। अनेन नूत्तसाहित्यनिर्माणं सम्पद्यते। साहित्यमिदं अन्यपरियोजनायाः आधारशिलायते इत्यस्मिन् नास्ति सन्देहलेशः।

उपसंहारः आधुनिकेऽस्मिन् काले संस्कृतक्षेत्रं ह्रासं गच्छति। तत्कारणानि सन्ति नैकानि। संस्कृतज्ञानां साङ्केतिकज्ञानाभावः संस्कृताध्ययनक्षेत्रे संस्कृतशोधक्षेत्रे च साङ्केतिकपरिज्ञानविनियोगाभावश्च संलक्ष्यते। अत एव संस्कृतक्षेत्रे उपाधेः अभावः जायते। तत्फलतः संस्कृताध्येतृणां संख्या प्रतिवर्षं क्षीयते। किन्तु संस्कृतवाङ्मयमेव सर्वविधविज्ञानस्य मूलभूतमस्ति। यन्त्रानुवादेन संस्कृतवाङ्मये निहितं विज्ञानसर्वस्वं सामान्यजनोपलभ्यं भवति चेत् पुनरेकवारं संस्कृतभाषायाः संस्कृतबाङ्मयस्य च पुनरुत्थानं नूनं भविष्यतीति।

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection

Digitized by Sarayu Trust Foundation Delhi and eGangotri.

CC0. In Public Domain Kirtikant Sharma Najafgarh Delhi Collection